





AN INFORMATION ORGANIZATION AND COLLABORATION TOOL FOR PROCESSING NOTES AND ACTION REQUESTS IN COMPUTER SYSTEMS

Patent number: WO9837474
Publication date: 1998-08-27
Inventor: ALLEN DAVID; SMIGA BRIAN; RABBANI DANNY;
 BUCHHEIM DENNIS; MANN TONY; HAGAN THOMAS;
 JOAQUIN JAMES
Applicant: ACTIONEER INC (US)
Classification:
 - International: G06F
 - european: G06Q10/00F
Application number: WO1998US02217 19980210
Priority number(s): US19970798453 19970210

Also published as:

 WO9837474 (A:
 EP1008080 (A3
 EP1008080 (A2
 US6026410 (A1

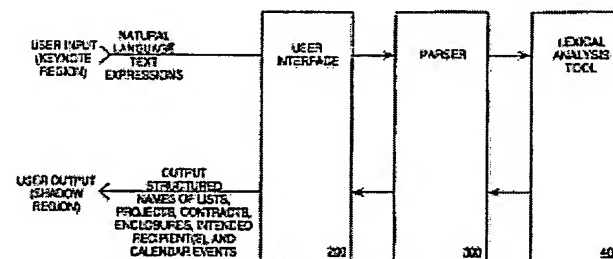
Cited documents:

 US5634121
 US5732274

Report a data error here

Abstract of WO9837474

A natural language-based information organization and collaboration tool for a computer system is disclosed. The present invention includes an apparatus and method for processing text expressions in a computer system, the apparatus including: 1) an object database defining an information object with an associated keyword; 2) a user input device for receiving an input text expression; 3) a parsing device for identifying the keyword in the input text expression, the parsing device including functions for linking the input text expression to the information object based on the keyword identified in the input text expression; and 4) a user output device for displaying to the user the identity of the information object to which the input text expression was linked. The apparatus of the present invention further includes supplemental information in the object database which is related to the information object, the user output device further including functions for displaying the supplemental information when a corresponding keyword is identified in the input text expression. The apparatus of the present invention further includes a method and apparatus for collaboration between users of a time and project management system.



Data supplied from the esp@cenet database - Worldwide

BEST AVAILABLE COPY

(19)日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11)特許出願公表番号
特表2001-512605
(P2001-512605A)

(43)公表日 平成13年 8月21日 (2001. 8. 21)

(51)Int.Cl.⁷

G 0 6 F 17/30

識別記号

1 8 0

F I

G 0 6 F 17/30

ターマコード* (参考)

1 8 0 E

1 8 0 D

審査請求 未請求 予備審査請求 有 (全116頁)

(21)出願番号 特願平10-536669
(86) (22)出願日 平成10年2月10日 (1998. 2. 10)
(85)翻訳文提出日 平成11年8月10日 (1999. 8. 10)
(86)国際出願番号 P C T / U S 9 8 / 0 2 2 1 7
(87)国際公開番号 W O 9 8 / 3 7 4 7 4
(87)国際公開日 平成10年8月27日 (1998. 8. 27)
(31)優先権主張番号 0 8 / 7 9 8 , 4 5 3
(32)優先日 平成9年2月10日 (1997. 2. 10)
(33)優先権主張国 米国 (U S)

(71)出願人 アクシオニア・インコーポレーテッド
アメリカ合衆国・94107・カリフォルニア
州・サンフランシスコ・プライアント ス
トリート・539
(72)発明者 アレン, デービッド
アメリカ合衆国・93023・カリフォルニア
州・オージェイ・マックニール ロード・
1674
(72)発明者 スミガ, プライアン
アメリカ合衆国・94107・カリフォルニア
州・サンフランシスコ・セカンド ストリ
ート・220番・461
(74)代理人 弁理士 山川 政樹 (外5名)

最終頁に続く

(54)【発明の名称】 コンピュータ・システムにおいてメモおよびアクション要求を処理する情報構成コラボレーシ
ョン・ツール

(57)【要約】

自然言語をベースとする、コンピュータ・システム用情報構成コラボレーション・ツールを開示する。本発明は、コンピュータ・システムにおいてテキスト表現を処理する装置および方法を含む。この装置は、1) 関連するキーワードを用いて情報オブジェクトを定義するオブジェクト・データベースと、2) 入力テキスト表現を受け取るユーザ入力装置と、3) 入力テキスト表現中の識別されたキーワードに基づいて入力テキスト表現を情報オブジェクトとリンクする機能を含む、入力テキスト表現中のキーワードを識別する解析装置と、4) 入力テキスト表現がリンクされた情報オブジェクトのIDをユーザに表示するユーザ出力装置とを含む。本発明の装置はさらに、情報オブジェクトに関する補助情報をオブジェクト・データベース内に含み、ユーザ出力装置はさらに、入力テキスト表現において対応するキーワードが識別されたときに補助情報を表示する機能を含む。本発明の装置はさらに、時間プロジェクト管理システムのユーザ同士の間のコラボレーション用の方法および装置を含む。

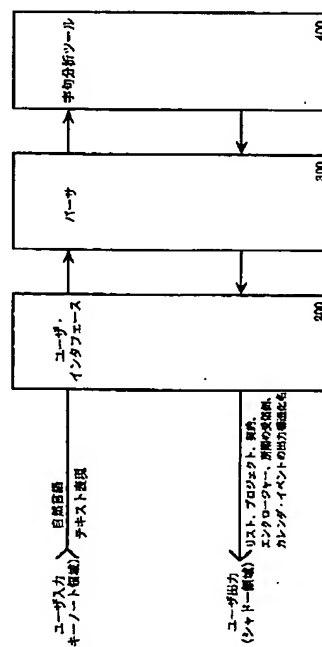


FIG. 2

【特許請求の範囲】

1. コンピュータ・システムにおけるテキスト表現を処理する装置であって、
関連するキーワードを用いて情報オブジェクトを定義するオブジェクト・データベースと、
入力テキスト表現を受け取るユーザ入力装置と、
入力テキスト表現中の識別されたキーワードに基づいて入力テキスト表現を情報オブジェクトとリンクする機能を含む、入力テキスト表現中のキーワードを識別する解析装置と、
入力テキスト表現がリンクされた情報オブジェクトの識別をユーザに表示するユーザ出力装置とを備えることを特徴とする装置。
2. オブジェクト・データベースが、情報オブジェクトに関する補助情報を含み、ユーザ出力装置がさらに、入力テキスト表現において対応するキーワードが識別されたときに補助情報を表示する機能を含むことを特徴とする請求項 1 に記載の装置。

【発明の詳細な説明】

コンピュータ・システムにおいてメモおよびアクション

要求を処理する情報構成コラボレーション・ツール

発明の分野

本発明は、コンピュータ・システムに記憶された情報の構成およびそのような情報へのアクセスに関する。詳細には、本発明は、構造化された情報出力を生成するための自然言語の分析と、コンピュータ・システムにおけるメモの処理に関する。本発明は、コンピュータ・システムを使用する時間およびアクション／プロジェクト管理にも関する。詳細には、本発明は、時間およびプロジェクト管理のための2人以上の人の間のコラボレーション（協働）用の方法および装置に関する。

関連技術の説明

従来技術には、情報を特定の方法で構成するか、あるいは特定の種類の情報を処理する多数のアプリケーション・プログラムが存在する。たとえば、文書処理（ワードプロセス）アプリケーションは特に、コンピュータ・システムでテキスト文書を処理するように構成されている。同様に、従来技術のデータベースは、明確に定義された方法でデータを構造化する手段を実現する。さらに、カレンダーリング・システムは、指定された日時に必要なイベントまたはアクションを追跡する構造化された方法を提供する。これらの従来技術のアプリケーションは情報を特定の方法で構成する方法を提供するが、多くの場合、ユーザが、情報を更新する必要があるときにアプリケーション・プログラム同士を切り換える必要があることは不都合である。さらに、このような種類のシステムでは、ユーザは、コンピュータ・システムにデータを入力する前に情報を構成しておく必要がある。たとえば、ユーザは、約束またはアクションの日付を入力する必要がある場合、カレンダーリング・プログラムを活動化するにはどうすべきかを知らなければならない。別々のアクションでは、ユーザは、カレンダーに入力した約束またはアクション

ョンに関連するリストまたはデータベースを更新する必要があることもある。多くの従来技術のシステムでは、ユーザは、ユーザ・インタフェースを扱い、情報

をそれに関連する所望のリストまたはカテゴリとリンクするのに時間を費やす必要がある。

「Information Management System」という名称の米国特許第5115504号は、データベースに記憶されている情報を表す要素同士をリンクするシステムについて説明している。このシステムは、要素から独立したデータベースのセクションに形成されたリンク構造と、第1の要素の位置を示すリンク構造内のポインタと、第2の要素の位置を示すリンク構造内の第2のポインタとを備える。データベースは、テキスト・データを有する項目と、各項目を複数のカテゴリとリンクできるように項目を分類することのできる複数のカテゴリとを備える。システムは、データベース内のある要素がある親カテゴリの子カテゴリに割り当てられている場合、その要素を自動的にその親カテゴリに割り当てる。システムは一般に、階層構成された複数のカテゴリのうちの1つまたは複数のカテゴリにデータベース内の要素を割り当てる手段を備える。システムは、カテゴリおよびセクション・ヘッドを有するセクションと、項目が割り当てられたカテゴリの所与のセクション・ヘッドの下方に順次表示される項目とに構成された情報画面としてビューを構築する。ユーザは、項目を入力した後、ビューのコラムに移動し、コラム・ヘッドの下にサブカテゴリの既存の名前を入力することによって、さらに直接割当てを行うことができる。このようにして、リンク構造が作成される。

残念なことに、米国特許第5115504号で開示されたシステムでは依然として、ユーザが表示画面上の情報カテゴリを直接処理する必要がある。この手法を使用する場合、ユーザは依然として、所望のカテゴリにデータを入力する際に情報を何らかの方法で構成する必要がある。多くの場合、ユーザが情報をこのようにして事前に構成し明示的に記憶することは非効率的であり不都合である。さらに、ユーザが好都合な自由形式または自然言語表現で入力を行うことができる場合には効率を高めることができる。ユーザが情報項目またはアクション項目を容易に、迅速に、かつユーザが現在使用している文脈を失わずに更新できる場合

にも従来技術のシステムが改善される。従来技術のシステムの他の欠点は、ユー

ザ同士またはアプリケーション同士の間のコラボレーション用の機構を提供しないことである。自分自身の情報を構成することは重要であるが、アクション／プロジェクトはしばしば、2人以上の人の間で共用される。言い換えれば、自分自身のToDoリストおよびカレンダーを構成するのでは不十分である。ユーザはまた、他のユーザと協働してプロジェクトを割り当て、プロジェクトの割当てを受け入れ、他のユーザに知らせることができるべきである。さらに、プロジェクトの状況を他のユーザに知らせることができると有利である。

複数のユーザが協働できるようにする従来技術の1つの方法は、eメールを使用して他のユーザにメッセージを送信することを含む。この従来技術のアプリケーションは、ユーザが任意の話題に関して通信することを可能にするが、ユーザによって維持されるカレンダーや、リストや、外部データベースにはリンクされない。したがって、eメールを交換するたびに合意が深まるにつれて、ユーザにはしばしば、プロジェクトに関連するあらゆることを更新する必要がある。ユーザが情報項目またはアクション項目を容易に、迅速に、かつユーザが現在使用している文脈を失わずに更新できる場合には、従来技術のシステムはさらに改善される。さらに、複数のユーザが協働し、多数の参加者の間でかつ多数のコンピュータ・システムを介してアクション項目を追跡することができるシステムを提供することによって、従来技術は改善される。

他の従来技術の方法はトップ・ダウン・デリゲーションを使用する。この方法では、スーパーバイザはプロジェクトを部下に委任することができる。しかし、この方法では、ユーザ同士の間の折衝やコラボレーションは不可能である。その代わりに、この方法では厳密な階層構造が使用される。したがって、この方法は、ピア同士の間のコラボレーションおよび折衝には有用ではない。このような従来技術のシステムの他の欠点は、クライアント・サーバ・システムを使用することである。すなわち、コラボレーション・プロセスに関する情報のレポジトリとしてサーバを設ける必要がある。したがって、コラボレーションに参加できるのは、サーバに接続されたクライアントだけである。これは、同じサーバとリンクされていない人は協働環境に参加できないことを意味する。

2人以上のユーザの間のコラボレーションを可能にするシステムを提供することによって、従来技術は改善される。さらに、複数のユーザが協働し、多数の参加者の間でかつ多数のコンピュータ・システムを介してアクション項目を追跡することができるシステムを提供することによって、従来技術は改善される。

したがって、より優れた自然言語情報構成コラボレーション・ツールが必要である。

発明の概要

本発明は、自然言語をベースとする、コンピュータ・システム用情報構成コラボレーション・ツールである。本発明は、コンピュータ・システムにおいてテキスト表現を処理する装置および方法を含む。この装置は、1) 関連するキーワード、プロジェクト、リスト、担当者、日時イベント、またはエンクロージャを用いて情報オブジェクトを定義するリレーショナル・オブジェクト・データベースと、2) 入力テキスト表現を受け取るユーザ入力装置と、3) 入力テキスト表現中の識別されたキーワードに基づいて入力テキスト表現を情報オブジェクトとリンクする機能を含む、入力テキスト表現中のキーワードを識別する解析装置と、4) 入力テキスト表現がリンクされた情報オブジェクトのIDをユーザに表示するユーザ出力装置とを含む。本発明の装置はさらに、情報オブジェクトに関する補助情報をオブジェクト・データベース内に含み、ユーザ出力装置はさらに、入力テキスト表現において対応するキーワードが識別されたときに補助情報を表示する機能を含む。本発明の装置はさらに、時間プロジェクト管理システムのユーザ同士の間のコラボレーション用の方法および装置を含む。

図面の簡単な説明

本発明を制限ではなく一例として添付の図面の図に示す。各図において、同じ参照符号は同じ要素を指す。

第1図は、本発明を実装することのできる完全なシステムを示す。

第2図は、本発明の主要システム要素のブロック図である。

第3図は、キーノート・シャドウ領域を示す表示画面を示す図である。

第4A図は、第4B図、第5図ないし第7図は、本発明のユーザ・インタフェ

ースの動作を示す図である。

第8図は、パーサの構成要素のブロック図である。

第9図は、字句分析ツールの処理フローを示す流れ図である。

第10図ないし第11図は、キーワード・パーサの処理フローを示す流れ図である。

第12図ないし第13図は、キーワード・パーサによって実行される処理の例を示す図である。

第14図は、提案リスト・キーワード機能の処理フローを示す流れ図である。

第15図ないし第18図は、好ましい実施形態のオブジェクト・データベース中のテーブルの構成を示す図である。

第19図は、クライアント・サーバ・システムの図である。

第20図は、ピア・ツー・ピア分散システムの図である。

第21図は、本発明の概要を示すフローチャートである。

第22A図は、発信側または要求側の見るコラボレーション・サイクルを示すフローチャートである。

第22B図は、受信側または代理人の見るコラボレーション・サイクルを示すフローチャートである。

第23図は、本発明の折衝プロセスを示すフローチャートである。

第24図は、FYIメモの分配を示すフローチャートである。

第25A図および第25B図は、コラボラティブ・プロセスの例を示す図である。

好ましい実施形態の詳細な説明

本発明は、自然言語をベースとする、コンピュータ・システム用情報構成コラボレーション・ツールである。以下の説明では、本発明を完全に理解いただくために多数の特定の詳細について述べる。しかし、当業者には、これらの特定の詳細なしに本発明を実施できることが明らかであろう。他の状況では、本発明を不必要にわかりにくくしないように周知の構造、回路、およびインタフェース

は詳しく示されていない。

第1図は、本発明の一実施形態が実装される典型的なデータ処理システムを示す。しかし、当業者には、様々なシステム・アーキテクチャの他の代替システムも使用できることが明らかになろう。第1図に示すデータ処理システムは、情報を伝達するバスまたは他の内部通信手段101と、情報を処理するためにバス101に結合されたプロセッサ102とを含む。システムはさらに、情報と、プロセッサ102によって実行すべき命令とを記憶するためにバス101に結合されたランダム・アクセス・メモリ（RAM）または他の揮発性記憶装置104（メイン・メモリと呼ぶ）を備える。メイン・メモリ104は、プロセッサ102によって命令が実行されている間に一時変数または他の中間情報を記憶することもできる。システムは、プロセッサ102に関する静的情報および命令を記憶するためにバス101に結合された読取り専用メモリ（ROM）および／または静的記憶装置106と、磁気ディスクや光ディスクなどのデータ記憶装置107およびそれに対応するディスク・ドライブも備える。データ記憶装置107は、情報および命令を記憶するためにバス101に結合されている。システムはさらに、コンピュータ・ユーザに情報を表示するためにバス103を介してバス101に結合された陰極線管（CRT）や液晶ディスプレイ（LCD）などの表示装置121に結合することができる。情報およびコマンド選択肢をプロセッサ102に伝達するために、英数字キーおよびその他のキーを含む英数字入力装置122をバス103を介してバス101に結合することもできる。他のユーザ入力装置には、方向情報およびコマンド選択肢をプロセッサ102に伝達し、表示装置121上のカーソルの移動を制御するために、バス103を介してバス101に結合されたマウス、トラックボール、スタイラス、カーソル方向キーなどのカーソル制御装置123がある。任意選択でバス103を介してバス101に結合できる他の装置には、ハード・コピー装置またはプリンタ124があり、このような装置を使用して、紙、フィルム、同様な種類の媒体などの媒体上に命令、データ、または他の情報を印刷することができる。好ましい実施形態では、ネットワークを介して分散システムの他のノードにアクセスする際に使用される通信装置125がバス103を介してバス101に結合される。通信装置125は、イーサネ

ット、トークン・リング、インターネット、またはワイド・エリア・ネットワークに結合するために使用されるようないくつかの市販のネットワーク化周辺装置を含むことができる。第1図に示すこのシステムのすべての構成要素および関連するハードウェアを本発明の様々な実施形態で使用できることに留意されたい。しかし、当業者には、特定の実施態様による様々な目的に任意の構成のシステムを使用できることが理解されよう。本発明の一実施形態では、第1図に示すデータ処理システムはIBM（登録商標）互換パーソナル・コンピュータまたはSun（登録商標）SPARCワークステーションである。プロセッサ102は、カリフォルニア州Santa ClaraのINTEL（登録商標）Corporationによって製造されている80486などの80X86互換マイクロプロセッサまたはPENTIUM（登録商標）ブランド・マイクロプロセッサのうちの1つでよい。

本発明を実装する制御論理またはソフトウェアは、メイン・メモリ104、大容量記憶装置107、またはプロセッサ102へローカルにアクセスすることのできる他の記憶媒体に記憶することができる。他の記憶媒体には、フロッピー・ディスク、メモリ・カード、フラッシュ・メモリ、またはCD-ROMドライブを含めることができる。当業者には、本明細書で説明する方法およびプロセスをメイン・メモリ104または読取り専用メモリ106に記憶されプロセッサ102によって実行されるソフトウェアとして実施できることが明らかであろう。この制御論理またはソフトウェアは、コンピュータ読取り可能媒体108を備える製造品上に存在することもでき、このコンピュータ読取り可能媒体108は、そこで実現され、大容量記憶装置107によって読み取ることのでき、プロセッサ102に本明細書の方法および教示に従って動作させるコンピュータ読取りプログラム・コードを有する。

本発明のソフトウェアは、前述のコンピュータ・ハードウェア構成要素のサブセットを含むハンドヘルド装置または携帯装置に実装することもできる。たとえば、ハンドヘルド装置は、バス101、プロセッサ102、およびメモリ104および／または106のみを含むように構成することができる。ハンドヘルド装置は、1組のボタンまたは入力信号構成要素を含むように構成することもでき、

ユーザはこれらのボタンまたは構成要素を用いて1組の利用可能なオプションのうちの1つを選択することができる。ハンドヘルド装置は、ハンドヘルド装置のユーザに情報を表示する液晶ディスプレイ（LCD）や表示要素マトリックスなどの出力装置を含むように構成することもできる。従来型の方法を使用してこのようなハンドヘルド装置を実現することができる。このような装置に関する本発明の実施態様は、本明細書で与えられる本発明の開示によって当業者に明らかになろう。

本発明は、自然言語をベースとする、解析可能で常に利用可能でインテリジェントなメモ・エディタであり、ユーザの考え、アクション要求、および情報をコンピュータ・システムに取り込むことができる。本発明は、ユーザから入力された自由形式のテキスト入力表現（本明細書ではキーノートと呼ぶ）におけるキーワードおよび日付情報を、自然言語解析を使用して識別し、識別された語に基づいて他の情報オブジェクトとのリンクを確立する。リンクされるこのような他のオブジェクトには、プロジェクト、担当者、日時イベント、リスト、および文書識別子オブジェクトが含まれる。キーワードとは、定義済みの一語テキスト文字列、または句読点を含むか、あるいは含まない定義済みの多重語テキスト文字列であり、1つまたは複数の関連情報オブジェクトに関連付けまたはリンクされる。リストとは、ユーザによって確立された、関連するキーノートの集合である。リストはアクション関連メモまたは単に保存されたメモでよい。本発明はさらに、アクション、メモ、個人キーノート、共用キーノート、アクション要求、FYI（for your information（情報））メッセージ、他のいくつかの異なる種類のキーノートのうちの1つなど特定の種類のキーノートとしてテキスト入力（すなわち、キーノート）を分類する。個人キーノートとは、他の誰にも送信されないキーノートである。共用キーノートは他人に送信される。キーノートが分類された後、本発明は、リンクされた担当者へアクション要求を送信するか、リンク・リスト、担当者、日時表現、またはプロジェクト・データを更新するか、コラボレーション・グループ内の他の人とキーノートを共有するか、あるいはキーノートに関する情報を整然と効率的に記憶することによって、キーノート进行处理する。本発明は、ユーザから入力テキスト表現を受け取り、キ

ーノートの分類に関する選択可能な補助情報をユーザに与えるリアルタイムな対話型ユーザ・インタフェースを含む。また、本発明は、前述のように自然言語を処理するパーサを含む。

第2図は、本発明の好ましい実施形態の主要構成要素をブロック図形で示す。ユーザは、ユーザ・インタフェース200に与えられるメモ、考え、またはアクション要求を表す自然言語テキスト表現（すなわち、キーノート）を与える。ユーザ・インタフェース200は、これらのテキスト表現をパーサ300に渡す。パーサ300は、キーノートの種類を識別し、入力テキスト表現内で見つけられた識別されたキーワードまたは日時表現に基づいて1つまたは複数の対応する情報オブジェクトにキーノートをリンクする責任を負う。リンクされたオブジェクトには、リスト、プロジェクト、担当者、eメール・アドレス、添付文書識別子、カレンダーで使用する日時イベントが含まれる。パーサ300は、字句分析ツール400を使用して、入力テキスト表現を複数のトークンに区分する。トークンとは、指定済みのデリミタ間の入力テキスト表現の一連の部分または互いに隣接する部分である。パーサ300がキーノートの種類を分類し、関連するオブジェクトにキーノートをリンクした後、リンク・リスト、プロジェクト、担当者、関連するeメール・アドレス、密閉文書識別子、任意のカレンダー・イベントがユーザ・インタフェース200に送り返され、表示装置121上のキーノート・シャドウ領域に表示される。

次に第3図を参照すると、ユーザ・インタフェース200のキーノート・シャドウ領域210の例が表示装置121の内容と対比して示されている。本発明のキーノート・シャドウ領域210は、その典型的なアプリケーションでは、他のウィンドウまたは表示装置121の情報領域および機能領域と組み合わせて表示装置121上に表示される。ウィンドウおよび情報領域または機能領域の表示装置上での表示は、当業者によく知られている。たとえば、ワシントン州RedmondのMicrosoft Corporationによって開発されたWindows 95™オペレーティング・システムは、そのようなウィンドウを表示するオペレーティング・システムの一例である。本発明のキーノート・シャドウ領域は、このようなオペレーティング・システムによって実現されるウィンドウ

機能および表示機能を使用して表示することができる。また、当業者には、このような情報領域を表示装置上に表示する他の手段を他の従来型のオペレーティング・システムまたはアプリケーション・プログラムでも実現できることが明らかであろう。当業者には、オペレーティング・システムの従来型のツールを使用して、キーノート・シャドウ領域 210 を任意の位置に、あるいは任意のサイズで表示できることも理解されよう。さらに、従来型のオペレーティング・システムは、表示優先順位または表示レベルを指定する手段を実現し、キーノート・シャドウ領域 210 を、常に最上表示優先順位または最上表示レベルで表示されるように指定またはコード化することができる。したがって、キーノート・シャドウ領域 210 は、従来型の方法を使用して、常に見ることができかつ常に表示画面 121 上で利用できるようにプログラムすることができる。このようにして、本発明は、入力されたテキストの解析およびキーワード一致のリアルタイム・フィードバックを行うウィンドウ領域または表示領域にテキスト情報を入力する常にご利用可能な方法を提供する。

次に、第 4 A 図および第 4 B 図を参照すると、好ましい実施形態のキーノート・シャドウ領域 210 の構成要素が例示されている。キーノート領域 220 は、ユーザが情報および考えを迅速に取り込むことを可能にする紙粘着メモのオン・スクリーン・バージョンである。キーノート領域 220 は、ユーザが取り込む必要のあるテキスト情報および考えを表す自然言語テキスト表現（すなわち、キーノート）が入力される表示領域を示す。キーノート領域 220 内のこのようなキーノートの例を第 4 B 図に示す。キーノート領域 220 には任意の種類の自然言語テキスト表現を入力することができる。従来型の技法を使用してテキストをキーノート領域 220 に表示しワード・ラップすることができる。さらに、従来型の技法を使用して、テキスト入力のエントリ用のキーノート領域 220 を選択または識別することができる。たとえば、カーソル制御装置 123、マウス、またはコンピュータ・システムのキーボード 122 上に入力された特殊キー・コードを使用して、テキスト表現が入力されるキーノート領域 220 を選択することができる。各キー・ストロークがキーノート領域 220 に入力されるにつれて、個々のキー・ストロークは、後述のようにユーザ・インタフェース 200 へ転送さ

れ、その後パーサ300へ転送される。

再び第4A図および第4B図を参照するとわかるように、解析されたテキスト表現の出力を表示し、ユーザから入力されたアイコン選択肢を取り込むシャドー領域230が用意される。シャドー領域230は、キーノート領域220の下、あるいはキーノート領域220に隣接して表示され、リンク・オブジェクト情報をデータ・フィールド内に含むウィンドウであり、このデータ・フィールドは、キーノート領域220に入力されたキーノートを解析した結果として自動的に設定される。シャドー230は1組のアイコン240を含み、これらのアイコンは、好ましい実施形態ではキーノート領域220と共に見える。アイコン240は2つの目的を果たす。第1に、アイコン240は、キーノートを解析した結果として、対応するオブジェクトとのリンクの有無を表す2つの状態のうちの1つで明確に表示することができる。たとえば、キーノートを解析した結果として担当者情報が見つかった場合、アイコン240のうちで、担当者情報オブジェクトを表すアイコンを強調表示することができる。同様に、キーノートを解析した結果として見つけれられるプロジェクト、リスト、カレンダー・イベント、密閉文書情報などのオブジェクトとのリンクの有無に応じて、プロジェクト、リスト、カレンダー、または密閉文書に対応する他のアイコンを強調表示または非強調表示することができる。別法として、プロジェクト・オブジェクト、リスト・オブジェクト、カレンダー・イベント・オブジェクト、密閉文書オブジェクトとリンクされたキーノートのキーワードをキーノート自体内に明確に表示することができる。たとえば、キーノートをプロジェクト・オブジェクトとリンクするキーワードを第1の色またはフォント・タイプまたは形式で表示することができる。キーノートを担当者オブジェクトとリンクするキーワードを第2の色またはフォント・タイプまたは形式で表示することができる。同様に、パーサ300が入力テキスト表現中の対応するキーワードを識別したことをユーザに知らせるように、キーノートを他のオブジェクトとリンクする他のキーワードを明確に表示することができる。

アイコン240および関連するドロップダウン・リスト制御によって実現される第2の機能は、ユーザが、選択された特定のアイコンに対応するリンク・オブ

ジェクト・タイプの表示を選択する手段である。パーサ300によって生成される出力は、シャドー230の領域250に表示される。シャドー230は最初、キーノート220の下に表示されるが、ユーザは、カーソル制御装置123を用いてシャドー領域230をクリックするか、あるいはキーボード122上の指定済みのキー・エントリをタイプすることによって、シャドー領域230をキーノート領域220の前方へ移すことができる。別法として、メニュー・コマンドを使用して、シャドー領域230を表示されるように選択することができる。

次に、第5図を参照すると、本発明の好ましい実施形態のキーノート・シャドー領域210の一般バージョンが初期状態で示されている。図のように、キーノート220は最初、任意のキーノートを入力しないうちは空白である。シャドー領域230は、全体的にI1ないしI5として識別された1組のアイコンを含む。この例では、アイコンI1はプロジェクト・オブジェクトを表し、アイコンI2は担当者アイコンを表し、アイコンI4は日時カレンダー・オブジェクトを表し、アイコンI5はリスト・オブジェクトを表す。当業者には、他の種類の情報またはオブジェクトがシャドー230の各アイコン240に対応できることが明らかであろう。同様に、当業者には、シャドー領域230に任意の数のアイコンを同様に設けられることが明らかであろう。

次に、第6図を参照すると、点線の円領域（図示のみのための領域）内に示されたキーノート222を入力した後のキーノート・シャドー領域が示されている。当業者には、点線が例示のみのために第6図に示されており、好ましい実施形態で実際には表示されないことが明らかであろう。キーノート222を解析した結果として、本発明によっていくつかの出力結果が生成されている。まず、パーサ300によって入力キーノート222とリンクされたオブジェクト・タイプに対応するアイコン240のうちの個々のアイコンが強調表示される。たとえば、パーサ300がプロジェクト・オブジェクト（すなわち、「wilson deal」（wilson取引））をキーノート222とリンクしているので、アイコンI1が強調表示される。同様に、パーサ300が担当者オブジェクト（すなわち、「Paul」）をキーノート222とリンクしているので、アイコンI2が強調表示される。パーサ300が日時カレンダー・イベント・オブジェクト（「

next Thursday」(来週の木曜日))を入力キーノート222とリンクしているので、アイコンI4が強調表示される。最後に、パーサ300がリスト(「Call」(電話))を入力キーノート222とリンクしているので、アイコンI5が強調表示されている。パーサ300が、特定のアイコンに対応する情報オブジェクト・タイプを入力キーノート222内で見つけることができない場合、シャドー領域230に、対応するアイコンが強調表示されることはないことが、当業者には明らかであろう。本発明によって入力キーノート222を解析した場合の第2の結果は、キーノートがアクション、メモ、個人キーノート、共用キーノート、アクション要求、FYIメッセージなど、いくつかの異なるキーノート・タイプの1つとして分類されることである。最後に、構造化出力情報またはリンク・オブジェクト・データがシャドー230の表示領域250に表示される。この出力情報については、第7図に関して詳しく説明する。

次に、第7図を参照すると、カーソル制御装置123または指定済みのキーボード122エントリを使用してシャドー領域230が前景に移された後のキーノート・シャドー領域210が示されている。シャドー230の領域250は、第6図に示すサンプル入力キーノート222を解析した結果として生成される構造化情報出力を示す。パーサ300は、入力キーノート222を解析した結果として、入力キーノート222中の「wilson deal」の参照を指定済みの「Wilson Account」プロジェクト・オブジェクトとリンクしている。リンクされたプロジェクト・オブジェクト「Wilson Account」は、対応するアイコンI1に隣接する領域250に表示される。同様に、パーサ300は、入力キーノート222中の「Paul」の参照を指定済みの担当者オブジェクト「Paul Jones」とリンクしている。リンクされた担当者オブジェクト「Paul Jones」は、対応するアイコンI2に隣接する領域250に表示される。パーサ300は、キーノート222中の「next Thursday」テキストを解析した結果として日時カレンダー・イベント・オブジェクトをリンクしている。処理されたこの日時カレンダー・イベント・オブジェクトは、シャドー230の、対応するアイコンI4に隣接する領域250に表示される。最後に、パーサ300は入力キーノート222中のキーワード「cal

1」を、リスト・オブジェクトとしてすでに定義されている指定済みの「C a l l s」リストとリンクしている。リンクされたC a l l s リストの識別は、シャドー230の、対応するアイコン15に隣接する領域250に表示される。

オブジェクトまたは情報の多重行リストが領域250のドロップ・ダウン部に表示されるように、リストを下向きに拡張させるドロップ・ダウン・インジケータ260が設けられる。コンピュータ表示装置上にドロップ・ダウン・リスト・インジケータを示す従来型の方法が存在する。

したがって、ユーザ・インタフェース200およびそれに対応するキーノート・シャドー領域210は、自然言語テキスト表現をユーザから受信し、テキスト表現の解析されリンクされた構造化出力を簡潔にかつ効率的に表示する手段および方法を実現する。以下に、パーサ300および字句分析ツール400によって実行される処理について詳しく説明する。

第3図ないし第7図と、本発明のユーザ・インタフェース200に関する上記の説明でわかるように、ユーザ・インタフェース200は、テキスト表現を入力し、結果として得られる関連する構造化情報を受信する容易で直感的なユーザ・インタフェースを実現する。さらに、キーノート・シャドー領域210が常に表示されるか、あるいは表示装置121上に容易に表示できるので、ユーザは、すでに行われた仕事の文脈を失わずにキーノート・ウィンドウ220内にメモまたは考えを容易に記録することができる。このようにして、本発明は、現在のユーザ操作を妨害せずにメモを容易に記録することを可能にする。また、本発明は、メモを自然言語非構造化形式で記録することを可能にする。この形式は、自然なユーザの思考プロセスに密に類似している。したがって、ユーザには、このようなメモや考えを特定の構造化フィールドに構成する必要がなく、かつユーザは、多重ステップ・アプリケーションを扱う必要なしにメモまたは考えを記録することができる。本発明の他の利点は、いくつかの従来型のコンピュータ・アプリケーションの動作を中央自由形式ユーザ・インタフェースに統合する能力である。本発明は、自然言語を、プロジェクト・オブジェクト、担当者オブジェクト、日時カレンダー・イベント・オブジェクト、またはリスト・オブジェクトとリンクされた構造化情報として解析する手段を提供するので、それによって生成された構

造化情報は、文書処理アプリケーション、カレンダーリング・アプリケーション、データベース・アプリケーション、プロジェクト管理アプリケーション、または電子メール・アプリケーションと容易に統合することができる。それによって、本発明は、ユーザが、構造化情報として解析することのできる非構造化テキスト表現を入力することを可能にし、この構造化情報はその後、このような様々な従来型のソフトウェア・アプリケーションに入力される。

パーサ

好ましい実施形態のパーサ300は、ユーザ・インタフェース200から自然言語テキスト表現を受け取り、入力テキスト表現中の識別されたキーワードに対応するプロジェクト、担当者、リスト、日時カレンダー項目、密閉文書などの情報オブジェクトとのリンクを含む構造化情報を生成する。従来技術には多数の解析アルゴリズムが存在するが、本発明のパーサ300は、入力テキスト文字列中のキーワードおよび／または日時カレンダー・イベントを有効に識別し提案し、対話型ユーザ・リアルタイム性能で応答する能力がユニークである。好ましい実施形態のパーサ300は、新規の内部アーキテクチャと、自然言語テキスト表現を処理する1組の方法とを用いてこれらの目的を達成する。以下に、本発明のパーサ300によって使用されるアーキテクチャおよび方法について説明する。

本発明は、自然言語テキストの構造および意味を解釈する問題を解決する。この意味は、ユーザに知られておりユーザによって事前に定義されている他の関連情報に関係するか、あるいはそのような情報にリンクされている1組の構造化情報である。以下の例は、本発明の動作を示す。

ユーザが、本発明のユーザ・インタフェース200に以下のサンプル・キーノートを入力するものと仮定する。

「call Scott tomorrow to arrange the next Engineering meeting (明日Scottに電話して次回のエンジニアリング会議の準備をする)」。ユーザがキーノートを入力するにつれて、本発明のパーサ300を使用してこのキーノートがリアルタイムで分析される。新しい各文字が入力された後にキーノート全体が解析

されることに留意されたい。キーノート全体がユーザによって入力されパーサ300によって分析された後、パーサ300によって以下の構造化情報出力が生成される。

```
lists:Calls  
  
project:arrange Engineering Meetings until Dennis gets back  
  
contact:Scott Jones  
  
date:tomorrow=current date+1day (リスト:電話  
プロジェクト:Dennisが戻るまでエンジニアリング会議の準備をする。  
担当者:Scott Jones  
日付:明日=現在の日付+1日)
```

この例では、本発明のパーサ300は、入力キーノート中のキーワード「calls」を認識し、このテキスト入力キーワードを「Calls」リストとリンクするか、あるいは関係付けるべきであると判定した。本発明のパーサ300は、キーワード「Scott」を認識し、この担当者名を担当者オブジェクト「Scott Jones」とリンクすべきであると判定した。「tomorrow」の語もパーサ300によって認識され、パーサ300は明日の日付を計算し（すなわち、現在の日付+1日）、この日付オブジェクトを入力キーノートとリンクした。キーワード「Engineering meeting」は、定義済みの「arrange Engineering meeting until Dennis gets back」プロジェクト・オブジェクトとのリンクとしてパーサ300によって認識された。本発明が、入力キーノートと対応するリンク・オブジェクト・タイプとを、表示領域250内のリスト・オブジェクト、プロジェクト・オブジェクト、担当者オブジェクト、または日時カレンダー・イベント・オブジェクト用のデータ領域に表示することに留意されたい。

本発明の1つの重要な目標は、入力された自然言語テキスト表現に基づいてコンピュータ・ユーザの意図を予想することによってユーザの作業量を軽減することである。ユーザの意図をかなり正確に予想することによって、本発明は、ユー

ザが情報をより効率的にかつ容易に取り込むことを可能にする。たとえば、上記で示した例の入力キーノート「call Scott tomorrow to arrange the next Engineering meeting」を単にタイプした後、ユーザは表示画面やアプリケーション・ユーザ・インタフェースを扱うことに貴重な時間を費やす必要なしに、入力キーノートを所望のリスト・オブジェクト、プロジェクト・オブジェクト、担当者オブジェクト、日時カレンダー・イベント・オブジェクトとリンクすることができる。本発明は、入力キーノート中の非構造化情報と対応する構造化情報オブジェクトとのリンクを自動的に処理する。したがって、ユーザは、キーノートの保存、キーノートの送信、またはキーノートに対するアクションの開始を非常に迅速に、ほとんどユーザの介入なしに行うことができる。

次に、第8図を参照すると、パーサ300の内部構造と、この内部構造とユーザ・インタフェース200、字句分析ツール400、およびオブジェクト・データベース850との関係がブロック図で示されている。パーサ300は、ユーザ・インタフェース200から入力された自然言語キーノートを受け取るキーワードと日時パーサ810を含む。キーワードと日時パーサ810は、キーワード・パーサおよび日時パーサを含む。キーワードと日時パーサ810のキーワード・パーサは入力キーノートからキーワードを解析する責任を負う。キーワードは、リスト、プロジェクト、担当者、文書密閉オブジェクト、および場合によっては日付（たとえば、「Dave's Birthday (Daveの誕生日)」= 6/25）を含め様々な異なるオブジェクト・タイプにリンクすることができる。これらの異なる種類のオブジェクトはそれぞれ、パーサ300に維持される。リスト・オブジェクト820は、ユーザによって定義されたリスト・オブジェクトを維持するために使用される。プロジェクト・オブジェクト822は、ユーザによって定義されたプロジェクト・オブジェクトを維持するために使用される。担当者オブジェクト824は、担当者名オブジェクトを維持するために使用される。エンクロージャ・オブジェクト826は、エンクロージャ・オブジェクトまたは文書識別子オブジェクトを維持するために使用される。他のオブジェクト・タイプ828も同様に設けることができる。各オブジェクト・タイプについて、

キーワードと日時パーサ810のキーワード・パーサは、以下に詳しく説明する方法でオブジェクト820ないし828を入力キーノートの対応するキーワードとリンクする。

キーワードと日時パーサ810の日時パーサは、日付イベントまたは時間イベントに対応する情報が存在するかどうかについて入力キーノートを走査するために使用される。日時パーサの動作については詳しく後述する。

キーワードと日時パーサ810は字句分析ツール400と相互接続される。当業者には、字句分析ツール400によって実行される機能をパーサ300またはキーワードと日時パーサ810の一体部分として実装することもできることが明らかであろう。しかし、本発明の好ましい実施形態では、字句分析ツール400はキーワードと日時パーサ810とは独立のソフトウェア・モジュールとして設けられる。キーワードと日時パーサ810は、ユーザ・インタフェース200から入力された自然言語キーノートを受け取ると、このキーノートを回線410上で字句分析ツール400に渡す。字句分析ツール400は、入力キーノートから1組のトークンを生成することに責任を負う。

次に、第9図を参照すると、字句分析ツール400によって実行される処理がフローチャートで表されている。最初の処理ブロック912で、字句分析ツール400がパーサ300から入力キーノートを得る。初期設定プロセスの一部として、文字ポインタがキーノートの第1の文字を指し示すように初期設定される。現在のトークンの文字を収集するために使用されるトークン・バッファは、論理ブロック912で初期設定される。次に、判断ブロック914で、入力キーノート中の各文字を処理するループが開始される。キーノート全体が処理されると、処理経路916が選択され、字句分析ツール400は出力トークン・リストをパーサ300に返す。入力キーノートにまだ文字が残っている場合、判断ブロック920への処理経路918が選択される。処理中の現在の文字がデリミタ文字ではない場合、処理ブロック928への処理経路924が選択され、処理ブロック928で、現在の文字がトークン・バッファに追加され、次の文字が論理ブロック930を介して処理され、判断ブロック914に戻る。デリミタ文字は、ブランク、タブ、コンマ、ピリオドなど定義済みの特殊文字のうちの1つでよい。再

び判断ブロック920を参照するとわかるように、現在の文字がデリミタ文字である場合、処理ブロック926への処理経路922が選択され、処理ブロック926で、現在収集されているトークンについて処理が完了する。論理ブロック926で、トークンを囲む空白または不要なブランクが除去され、トークンの大文字または小文字を任意選択で、一貫した大文字形式に標準化することができる。さらに、トークンは、 α 文字、 α 数字、日付、日付スパンなど、いくつかのトークン・タイプのうちの1つとして分類される。数量もトークンに関連付けられる。トークンはまた、日時トークンまたはキーワード・トークンとして分類される。好ましい実施形態では、キーワード・トークン用のトークン・リストおよび日時トークン用の別のリストの2つの別々のトークン・リストが維持される。このように区別するのは、パーサ300が文脈の影響を受けるからである。たとえば、トークン「Friday（金曜日）」は、日時パーサには特殊な意味を有するが、キーワード・パーサには有さない。当業者には、代替実施形態では別々のトークン・リストを使用することが不要であることが明らかであろう。

次いで、新しいトークン、トークン・タイプ、トークン値情報が出力トークン・リストに追加される。出力トークン・リストは、字句分析ツール400によって実行される処理が完了したときにパーサ300に返されるトークン・リストを表す。やはり処理ブロック926で、トークン・バッファが、次のトークンを収集できるようにセットアップするように初期設定され、処理が処理ブロック930へ転送され、キーノート中の次の文字が処理される。本発明は、第9図に示すように字句分析ツールによって実行されるこの基本処理フローを使用して、自然言語自由形式入力キーノートを、パーサ300によって好都合に処理できる1組のトークンおよびトークン・タイプ・トークン値情報に変換する。このトークン・リストは、第8図に示すように経路420上で字句分析ツール400からパーサ300へ転送される。

パーサ300と字句分析ツール400との間のインタフェースを使用してパーサ300と字句分析ツール400との間でトークンを交換し、特定の日時イベントを表すために使用できる多重語トークンを得ることもできる。パーサ300と字句分析ツール400は、パーサ300が、日時トークンの後に続く日時処理経

路を見つけるまで日時トークンを交換する。

第10図ないし第13図は、本発明のキーワードと日時パーサ810のキーワード・パーサの動作を示す。第10図および第11図は、キーワードと日時パーサ810のキーワード・パーサの処理フローを示すフローチャートである。第12図および第13図は、キーワードと日時パーサ810のキーワード・パーサの動作の例を示す。

次に第12図を参照すると、最初にオブジェクト・ディレクショナリ851にキーワード情報をロードする方法が例示されている。初期設定時には、オブジェクト・データベース850がアクセスされ、オブジェクト・ディクショナリ851にロードすべき初期オブジェクトが得られる。オブジェクト・ディクショナリ851内のローカル・データを使用できるので、パーサ300の動作はずっと高速である。オブジェクト・データベース850については詳しく後述する。

オブジェクト・ディクショナリ851は、トリガ・テーブル856、キーワード・ディクショナリ852、キーワード定義テーブル854を含む。トリガ・テーブル856は、キーワードが形成される各トークン用のトリガと呼ばれるエントリを含む。各トリガには、キーワードの数を識別する基準カウントが関連付けられ、対応するトリガはこれらのキーワードの一員である。キーワード・ディクショナリ852は、キーワードと日時パーサ810のキーワード・パーサによって実行されるメソッドまたは呼出しを使用して事前に定義された各キーワードのIDを含む。第12図に示す論理ブロック1210、1212、および1214は、キーワード・ディクショナリ852にエントリを追加するために使用される「キーワード追加」関数または「キーワード追加」メソッドを呼び出す3つの例を示す。このような呼出しは、キーワード定義テーブル854にキーワード定義情報をロードする場合にも使用される。第12図の例に示すように、キーワード定義は、キーワード・ディクショナリ852中の各キーワードに関連するリスト・オブジェクト、プロジェクト・オブジェクト、担当者オブジェクトの参照を含む。キーワード・ディクショナリ852は定義リンク(Def. Link)を含み、この定義リンクは、キーワード・ディクショナリ852中のキーワードに対応するキーワード定義エントリを指し示す。第12図に示すサンプル・メソッド

呼出し1210、1212、および1214の結果として、キーワード・ディクショナリ852、キーワード定義テーブル854、およびトリガ・テーブル856に、キーワードと、関連するキーワード定義およびトリガとが含まれる。オブジェクト・ディクショナリ851のこれらのテーブルおよび記憶領域が含まれた後、キーワードと日時パーサ810のキーワード・パーサを使用して、ユーザ入力キーノート、これらの定義済みキーワードを含む自然言語から出力構造化情報テーブルとして解析することができる。この出力構造化情報テーブルは、キーワードと日時パーサ810のキーワード・パーサによって検出されたユーザ入力キーノート中のキーワードに関連するリンク・リスト・オブジェクト、プロジェクト・オブジェクト、および担当者オブジェクトを含む。

次に第10図および第11図を参照すると、この機能を実行するためにキーワードと日時パーサ810のキーワード・パーサによって使用される処理論理がフローチャートで示されている。初期処理ブロック1010で、キーワード・バッファ、現キーワード・バッファ、および出力バッファが初期設定される。キーワード・バッファは、最も完全な（すなわち、最長の）キーワードが構築できるまでキーワードの一部（すなわち、1つまたは複数のトークン）を一時的に保持するために使用される。現キーワード・バッファは、現在最も完全なキーワードを記憶するために使用される。出力バッファは、現在のユーザ入力キーノートのキーワードに関して検索された構造化情報またはキーワード定義情報を記憶するために使用される。処理ブロック1012で、ユーザ入力キーノート中の次のトークンが検索される。検索されたトークンが、トリガ・テーブル856にアクセスすることによって判定されたトリガである場合、処理ブロック1022への処理経路1018が選択され、処理ブロック1022で、キーワード・バッファの内容にトリガが付加される。次に、判断ブロック1024で、キーワード・ディクショナリ852でキーワード・バッファの現在の内容が探索される。キーワード・ディクショナリ852でキーワード・バッファの内容が見つかった場合、処理は処理ブロック1026に進み、キーワード・バッファの内容が現キーワード・バッファに記憶される。この場合、キーワード・ディクショナリ852でキーワード・バッファ中のトークンの現在の集合が見つかった。ユーザ入力キーノ

ート中にまだトークンがある場合、処理ブロック1012に戻る経路1030が選択され、処理ブロック1012で次のトークンが検索され、上記の処理が繰り返される。

再び判断ブロック1014を参照するとわかるように、検索されたトークンが、トリガ・テーブル856にアクセスすることによって判定されたトリガではない場合、判断ブロック1020への処理経路1016が選択され、判断ブロック1020で、現キーワード・バッファの内容が検査される。現キーワード・バッファが空である場合、処理は、Aで示されたバブルを介して判断ブロック1028に進む。この場合、現在のトークンは単に破棄される。しかし、現キーワード・バッファが空でない場合、処理は、第11図に示すBで示されたバブルに進む。同様に、ユーザ入力キーノート中の最後のトークンについて現キーワード・バッファが空でなく、したがって処理経路1034が横断される場合、処理は、Bで示されたバブルに進む。

次に第11図を参照するとわかるように、Bで示されたバブルでキーワードと日時パーサ810のキーワード・パーサに関する処理が行われる。この場合、現キーワード・バッファは、キーワード・ディクショナリ852中の定義済みのキーワードを形成する、ユーザ入力キーノートで見つけられる最大数の連続トークンを含む。この場合、現キーワード・バッファ中の現在のキーワードは、キーワード・ディクショナリ852中の関連するキーワードを参照するために使用される（処理ブロック1110）。キーワード・ディクショナリ852でキーワードが見つかった後、キーワード定義テーブル854から対応するキーワード定義が検索される。対応するキーワード定義には、キーワード・ディクショナリから得た一致したキーワードに対応するリスト・オブジェクト情報、プロジェクト・オブジェクト情報、および担当者オブジェクト情報が含まれる（処理ブロック1112）。判断ブロック1114で、出力バッファ内でキーワード定義構成要素（すなわち、リスト、プロジェクト、および担当者）がすでに修正されているかどうかを判定する試験が実行される。これが真である場合（処理経路1118）、処理ブロック1116の動作を実行することによって出力バッファの現在の内容がさらに修正されることはない。処理ブロック1116で、キーワード・ディク

ショナリからのキーワード定義情報が出力バッファの未修正部分に記憶される。

本発明では、キーワードやキーワードのトークンが混合されておらず、キーワードと日時パーサ810のキーワード・パーサが、それぞれの異なるキーワード定義とリンクされた2つ以上のキーワードを検出した場合、好ましい実施形態のキーワードと日時パーサ810のキーワード・パーサは、キーワードがユーザ入力キーノートの最初または最も左に表示されたキーワード定義を選択する。さらに、2つ以上のキーワードまたはキーワードのトークンが混合されている場合、好ましい実施形態のキーワードと日時パーサ810のキーワード・パーサは、ユーザ入力キーノート中の左から右への最長のキーワードを選択し、次いでこの最長のキーワードのすべてのトークンを破棄し、ユーザ入力キーノートの残りの部分进行处理する際に考慮されないようにする。本発明におけるこれらの解析規則を使用すれば、入力キーノートが与えられた場合に出力構造化情報を予測し理解することができる。出力バッファが処理ブロック1116にロードされた後、処理は、第10図に示すAで示されたバブルに進む。再び第10図と、Aで示されたバブルを参照するとわかるように、トークン进行处理する処理ループは、入力キーノート中のすべてのトークンが処理されるまで反復的に判断ブロック1028に進む。

次に第13図を参照すると、本発明のキーワードと日時パーサ810のキーワード・パーサによって生成される対応する出力構造化情報に関連するユーザ入力キーノートが与えられた場合の、本発明の好ましい実施形態の動作として4つの例（1ないし4）が示されている。さらに、第13図に示す例は、第12図に示すロード済みオブジェクト・ディクショナリ851に基づく例である。上記で第12図に関して説明したように、トリガ・テーブル856、キーワード・ディクショナリ852、およびキーワード定義テーブル854を含むオブジェクト・ディクショナリ851には、第12図に示すサンプル情報などの情報がロードされる。この情報に基づいて、本発明は、各例ごとに示された入力キーノートが与えられた場合、第13図に示す出力構造化情報を生成する。

次に第12図および第13図を参照するとわかるように、第1の例では、入力キーノート「hello world」が与えられた場合、本発明はこの入力キ

ーノートを、第12図に示すキーワード・ディクショナリ852中の行2にある「hello world」キーワードと突き合わせる。その結果、第12図に示すキーワード定義テーブル854の行2の内容が、第13図の第1の例に示す出力構造化情報バッファへ転送される。

第13図に示す第2の例では、入力キーノート「hello world peace」が本発明によって解析される。その結果、本発明はこの第2のサンプル入力キーノートを、キーワード・ディクショナリ852中の行2にある「hello world」キーワードと突き合わせる。この場合、第12図に示すキーワード定義テーブル854の行2にあるキーワード定義が、第13図の第2の例に関して示すように出力構造化情報バッファへ転送される。第13図に示す第2の例のキーワード「world peace」によって出力構造化情報中のキーワード定義が修正されることがないことに留意されたい。

次に、第13図に示す第3の例を参照するとわかるように、入力キーノート「hello everyone world peace」がキーワードと日時パーサ810のキーワード・パーサによって処理される。この場合、キーワードと日時パーサ810のキーワード・パーサは、「hello」キーワードを、第12図に示すキーワード・ディクショナリ852の第1行と突き合わせる。この場合、第12図に示すキーワード定義テーブル854の行1にあるキーワード定義が、第13図に示す第3の例に関して示された出力構造化状態バッファへ転送される。第13図に示す第3の例の入力キーノートをさらに解析する際、キーワード「world peace」が、第12図に示すキーワード・ディクショナリ852の第3行と突き合わせられる。しかし、キーワード定義854のリスト構成要素および担当者構成要素は、すでに一致したキーワード（「hello」）によってすでに修正されているので、出力構造化情報中のリスト構成要素および担当者構成要素が「world peace」キーワード定義に関してさらに修正されることはない。

第13図に示す第4の例を参照するとわかるように、入力キーノートはキーワードと日時パーサ810のキーワード・パーサによって解析される。この場合、キーワード「hello」がキーワード・ディクショナリ852の第1行と一致

し、第13図に示す第4の例に関して示されたように、行1の対応するキーワード定義が出力構造化情報バッファへ転送される。しかし、この場合、第4の例で入力キーノートをさらに解析すると、キーワード「hello world」がキーワード・ディクショナリ852の第2行と突き合わされる。行2から対応するキーワード定義が得られる。キーワード定義のプロジェクト構成要素はまだ修正されていないので、第13図に示す第4の例に関して示された出力構造化情報バッファのプロジェクト構成要素へキーワード定義テーブル854の第2行のプロジェクト定義を転送することができる。

本発明は、定義済みのオブジェクト・ディクショナリ851および入力ユーザ・キーノートが与えられた場合にユーザに新しいキーワードを提案するためのメソッドおよびオブジェクトを含む。

次に第14図を参照すると、担当者オブジェクトにリンクすべきキーワードを提案するために実行される処理を示すフローチャートである。当業者には、リスト、プロジェクト、エンクロージャ、またはその他のキーワード定義構成要素に関して同様な1組の処理を実行できることが明らかであろう。処理ブロック1410で、字句分析ツール400によってユーザ入力キーノートがトークンとして解析される。次いで、各トークンをキーワード・ディクショナリ852と比較し、トークンがすでに定義済みのキーワードであるかどうかを判定することができる（処理ブロック1414）。また、提案できない語として明示的に定義された語の定義済みリストの内容を用いてトークンを検査することができる。トークンがまだキーワード・ディクショナリ852になく、かつ提案できない語のリストにない場合は、このトークンを処理ブロック1416でキーワードとして提案することができる。トークンがすでにキーワード・ディクショナリ852内にあるか、あるいは提案できない語のリスト上にある場合は、このトークンを提案することはできない。この場合、トークンをキーワード・ディクショナリ852内に現在存在している他のキーワードと明確に区別するように様々な方法で補足することができる。トークンのこのような補足の一例として、入力キーノート中の次の順次トークンの第1の文字を使用し、現在処理中のトークンと連結することができる。次いで、この補足されたトークンをキーワード・ディクショナリ852

の内容と比較し、補足されたトークンが現在、キーワード・ディクショナリ内にも、提案できない語のリスト上にも存在しないかどうかを判定することができる。補足されたトークンが、キーワード・ディクショナリ852内か、あるいは提案できない語のリスト上で見つかった場合、入力キーノート中の次のトークンまたは前のトークンの追加の文字を使用して、補足されたトークンをさらに補足することができる。このプロセスは、固有の提案できる補足されたトークンが生成されるまで継続する。次いで処理ブロック1418で、この固有の提案できる補足されたトークンをユーザにキーワードとして提案することができる。入力キーノート中のすべてのトークンを組合せトークンとして互いに付加することができる。この組合せトークンは、キーワード・ディクショナリ852の内容および提案できない語のリストと比較され、組合せトークンが現在キーワード・ディクショナリ中のキーワードとして定義されているか、それとも提案できない語として定義されているかが判定される。そうでない場合、処理ブロック1420で、組合せトークンを追加のキーワードとして提案することができる。

代替実施形態では、キーワードを提案する論理は、キーワードが提案されているオブジェクトの種類に基づいてそれぞれの異なるヒューリスティックを使用することができる。したがって、たとえば、上記で担当者オブジェクトに関して説明したように、後続のトークンの文字を含めるようにキーワードを補足するか、あるいは担当者の名前の場合と同様に複数のトークンからイニシャルを作成することが望ましいことがある。しかし、このプロセスは、リスト・オブジェクトまたはプロジェクト・オブジェクト用の補足されたキーワードを作成する場合に適切ではない恐れがある。このような場合、いくらか異なる補足プロセスを使用することができる。提案プロセスが使用される時点ではオブジェクトの種類がわかっているため、オブジェクトの種類に基づくそれぞれの異なるヒューリスティックの使用は、本発明を用いて容易に実現される。キーワード提案が生成されているオブジェクトの種類を知ることによって、適切なヒューリスティックを選択することができる。

したがって、キーワードを解析するために本発明によって実行される処理について説明する。

日時パーサ

以下に、本発明のキーワードと日時パーサ810の日時パーサの実施態様について説明する。キーワードと日時パーサ810の日時パーサは字句分析ツール400を使用して、入力キーノートの日付関連トークンに分解し、日付解析に関連する入力表現中のトークンを識別する。字句分析ツール400は、上記で第9図に関して説明した方法と同様な、入力キーノートの日付関連トークンに分解する方法を使用する。字句分析ツール400は、各トークンにそのトークンの種類の指定およびそのトークンに関連する数値を与える。たとえば、語「2」は、2に等しい値を有する数値タイプ・トークンとして字句分析ツール400によって分類される。他の例として、トークン「Monday」は、週の第2日に対応する2に等しい数値を有する日タイプ・トークンとして字句分析ツール400によって分類される。他の例として、トークン「/」は、不定数値を有するデータ分離子トークンとして分類される。同様に、日付イベントまたは時間イベントに関連する他のすべてのトークンが、特定のタイプおよび定義済みの数値を用いて事前に定義される。このようなタイプおよび数値は、入力キーノート中の特定のトークンを識別する際にキーワードと日時パーサ810への出力として字句分析ツール400から与えられる。キーワードと日時パーサ810の日時パーサは、字句分析ツール400から与えられるトークンおよびトークン・タイプ情報を使用して、入力キーノート中の日時イベントを解析し認識することができる。

キーワードと日時パーサ810の日時パーサは、入力キーノートを解析し、ユーザから入力された自然言語入力キーノートから日付および時間を解釈する責任を負う。本発明の日時パーサ814の目標は、入力キーノートから1組の日付トークンおよび時間トークンを解析し、ユーザによって適用される所期の日付制約または時間制約を完全に正確に判定することである。本発明によって使用されるフォーマットは、ユーザが、迷路のような対話に対処するか、あるいは多数のコマンド・ボタン、チェック・ボックス、またはその他のグラフィカル・ユーザ・インタフェース構成要素を操作することを強制されるのではなく、短く簡単な表現を自然言語として直接タイプすることによって、ほぼあらゆる日時値を表現す

ることを可能にするほど簡単で、短く、直感的かつ効果的である。

キーワードと日時パーサ 810 の日時パーサによって実行される日時解析は、新規の再帰的下降形態を無限ルック・アヘッド技法と共に使用する。この技法は、 $O(N)$ 複雑さを実現する。同様に、キーワードと日時パーサ 810 の日時パーサも $O(N)$ 複雑さを達成する。これは、理論的に可能な最高の複雑さである。したがって、本発明の性能は、高速の応答を必要とするリアルタイム・アプリケーションによく適している。

以下に、認識される日付表現のリストと、好ましい実施形態のキーワードと日時パーサ 810 によって使用される規則のリストを示す。

注意：この表を読んでいる間、今日は 1996 年 6 月 10 日月曜日であると仮定する。

<u>Date Description (日付の説明)</u>	<u>Parses to This Date (この日付に解析)</u>
---------------------------------	--------------------------------------

• today (今日)	Mon., 6/10/96 (1996 年 6 月 10 日月曜日)
• this morning/afternoon/evening (今日の午前／午後／夕方)	Rule: Parse to current date (規則：現在の日付に解析)
• tonight (今夜)	
• tonite	

• tomorrow (明日)	Tues., 6/11/96 (1996 年 6 月 11 日火曜日)
• tomorrow morning/afternoon/evening/night (明日の午前／午後／夕方／夜)	

Rule: Parse to the current date + 1 day
(規則：現在の日付 + 1 日に解析)

・ [0]6/10

Mon., 6/10/96

(1996年6月10日月曜日)

・ [0]6-10

Rule: Parse to the current date or the nearest future occurrence of the specified date. (規則: 現在の日付または指定された日付の最も近い将来での発生に解析する。)

・ [0]6.10

(6月10日)

・ June 10

・ June 10th

・ June ten[th]

・ ten[th] of June

・ [0]6/10/98

Wed., 6/10/98

(1998年6月10日水曜日)

・ [0]6-10-98

Rule: Parse to the specified date.

(規則: 指定された日付に解析)

・ [0]6.10.98

Note: The parser must be configured for European locations to enable handling of the DD/MM/YY date format. (注意: パーサは、DD/MM/YY日付フォーマットの処理を可能にするようにヨーロッパ・ロケーション向けに構成しなければならない。)

・ June 10, 1998

(1998年6月10日)

・ 10 June 1998

・ tow days* from today**

Wed., 6/12/96

(今日から2日)

(1996年6月12日)

Rule: Parse to the Specified starting date ("today," etc.) + specified increment ("two days," etc.) (規則: 指定された開始日 ("今日" など) + 指定された増分 ("2日" など))

• in/within fourteen days*

Mon., 6/24/96

(14日内)

(1996年6月24日月曜日)

Rule: Parse to the current date +

specified number of days or weeks. Adding months will yield the same day of the month (if possible), x months later. In other words, "2 months from "june 4th, 96" will yield "August 4th, 96", "one month from May 31st", will yield "June 30th" (the last day of the month, since June 31st does not exist). Similarly, "five years from 5/5/94 " will yield "5/5/99". (規則: 現在の日付+指定された数の日または週。月を加えると、(可能なら) xカ月後の同じ日が生成される。言い換えれば、「2 months from June 4th, 96 (1996年6月4日から2カ月)」は、「August 4th, 96 (1996年8月4日)」を生成する。「one month from May 31st (5月31日から1カ月)」は「June 30th (6月30日)」(6月31日は存在しないので月の最後になる)。同様に、「five years from 5/5/94 (1994年5月5日から5年)」は「5/5/99 (1999年5月5日)」を生成する。

• Monday

Mon., 6/17/96

(月曜日)

(1996年6月17日月曜日)

• this Monday (今度の月曜日)

Rule: Parse to next occurrence of the

specified day; never the current date. For example, Tuesday" is Tue, 6/11/96. (規則: 指定された日の次の発生に解析。現在の日付ではない。たとえば、「Tuesday (火曜日)」は1996年6月11日火曜日である。)

• this coming Monday (来週の月曜日)

• next Monday

Mon., 6/17/96

(次の月曜日)

(1996年6月17日月曜日)

Rule: Parse to: (規則:)

1. "next<day>" is specified during the next week, using Monday as the first day of the week. (1. 「next<day> (次の日)」は、月曜日を週の第1日として使用して、次の週の間に指定される。)

2. "On a Sunday, "next<day>" parses to the specified day during the 2nd week following the Sunday (i.e., the week beginning eight days from the Sunday)-except for Sunday, which parses to the Sunday that is one week away. (2. 「On a Sunday, next<day>」は、日曜日の後に続く2番目の週（すなわち、日曜日から8日目に始まる週）中の指定された日に解析される。この場合、日曜日自体は除かれ、日曜日は1週間先の日曜日に解析される。)

・ first*** Monday of next month Mon., 7/3/96
 (来月の第1*** 月曜日) (1996年7月3日月曜日)

・ first*** Monday of/in July Rule: Parse to the first occurrence of
 the specified day during the next month on the calendar.
 (7月の第1*** 月曜日) (規則: カレンダー上の次の月の間の指定され
 た日の発生に解析される。)

・ first*** Monday of June, 1995
 ・ first*** Monday of June, 95
 ・ first*** Monday of 6/95
 (1995年6月の第1*** 月曜日)

* 「日」ではなく「週」、「月」、「年」、「月曜日」、「火曜日」などを指定することができる。

** 「今日」ではなく「明日」、「来週の火曜日」などを指定することができる。

*** 「第2」、「第3」、「第4」、「第5」、および「最後」を指定すること

ができる。

再発イベントは、日、週、月、年単位で発生することができる。キーワードと

日時パーサ810は再発イベントのエントリも処理する。キーワードと日時パーサ810は、以下の種類の再発イベント指定および関連する日付関連キーワードまたはキー表現を認識する。他の形式を同様に処理できることに留意されたい。

- ・ 毎日*
 - ・ 1日おき*
 - ・ 3日おき*
 - ・ 毎木曜日
 - ・ 週に1度（毎月曜日、毎火曜日、毎水曜日、毎木曜日、毎金曜日）
 - ・ 毎週末（毎土曜日および毎日曜日）
 - ・ 隔週末
 - ・ 隔週の木曜日
 - ・ 毎火曜日
 - ・ 毎週末
 - ・ 毎月（1カ月おき）の第1木曜日
 - ・ 3カ月おきの第1木曜日
 - ・ 毎月の第15日
- * 「日」は、「週」、「月」、および「年」で置き換えることができる。

好ましい実施形態では再発日付が永遠に発生するものと仮定されることに留意されたい。キーワードと日時パーサ810は常に、適切な開始日を選択するが、終了日付は報告しない。たとえば、今日が1996年6月11日であると仮定する場合、「毎火曜日」は1996年6月12日火曜日（今日から最も近い週の火曜日）から始まる「毎火曜日」としてパーサ810によって解釈される。

日付と同様に、略号および数値／序数代用符号も正しく解釈される。

本発明の好ましい実施形態のDLLインタフェース

以下に、本発明の好ましい実施形態のパーサ300ダイナミック・リンク・ライブラリ（DLL）構成要素とのインタフェースについて説明する。DLLの一般的な用途は当業者によく知られている。

好ましい実施形態のパーサ300は単一のDLLで構成される。DLLはC＋

+プログラミング言語で書かれる。当業者には、C、Basicなど他のプログラミング言語を使用することもできることが明らかであろう。好ましい実施形態のパーサ300機能には以下の機能が含まれる。

- ・日時解析
- ・再発イベント解析
- ・キーワードを介したリスト、プロジェクト、および担当者の解析
- ・リスト・キーワード、プロジェクト・キーワード、および担当者キーワード

の提案

・リスト名、プロジェクト名、および担当者名の自動完成（自動完成または自動充填）。このプロセスは、すでに入力されたデータに基づいて部分的データ・エントリを完成することによってユーザを助けるように設計された従来型の「クイック・フィル」技法と類似している。

- ・コラボレーション解析

パーサ300の観点からは、日時解析にデータの特種な知識は必要とされない。一方、キーワード、リスト、プロジェクト、および担当者の解析、完成、および提案は、既存または定義済みのリスト、プロジェクト、担当者、およびキーワードの明示的な知識を必要とする。好ましい実施形態のパーサ300はデータに依存しない。言い換えれば、パーサ300はファイルやデータベースを認識しない。したがって、パーサ300をリスト、プロジェクト、担当者、およびキーワードで初期設定しなければならない。この初期設定プロセスは通常、パーサ300を使用するアプリケーションのブート時、あるいはアプリケーションが他の1組のデータに切り換えるときに行われる。以下の擬似コードは、好ましい実施形態のパーサ300の典型的な初期設定を示す。

```
declare list, project, contact, keyword as strings
for every list in database
```



```
AddList(list)
```

```
for every project in object database
```

```
AddProject(project)
```

```
for every contact in object database
```

```
AddContact(contact)
```

```
for every keyword in object database
```

```
AddKeyword(keyword, list, project, contact)
```

この初期設定の後、パーサ300はすべてのリスト、プロジェクト、担当者、およびキーワードを知る。この場合、パーサ300は、クライアントからの要求に応じてキーワードを自動充填し、解析し、提案することができる。もちろん、パーサ300はオブジェクト・データベース850内のデータとの同期を維持しなければならない、オブジェクト・データベース850の変化はパーサ300に反映されるべきである。更新は、Add関数呼出し、Delete関数呼出し、およびRename関数呼出しを使用して行われる。一例として、以下の場合を検討されたい。ユーザが「Paint Fence」という名前の既存のプロジェクトを削除する。アプリケーションは、オブジェクト・データベース850からこのプロジェクトを削除し、それに関連するキーワードを削除（または更新）する。この変更は、パーサ300に反映されなければならない、以下の単一の関数呼出しを用いて行うことができる。

```
DeleteProject("Paint Fence");
```

この単一の関数呼出しは、プロジェクトおよびその参照をパーサ300から削除する。もはやこのプロジェクト名が自動的に完成されることはなく、「Paint Fence」プロジェクトに関連するすべてのキーワードが自動的に削除または更新される。DeleteProject()、AddProject()、およびRenameProject()が、関数の成功または失敗を示す値

を返すことに留意されたい。話を簡単にするために、前の例では返却値を無視する。

R e n a m e 関数は、残りのリスト、プロジェクト、担当者、およびキーワードをサポートする。リスト（「リスト」は一例として使用されており、「プロジェクト」または「担当者」で置き換えることができる）の名前変更はオブジェクト・データベース850で容易に行われる。これは實際上、簡単なデータベース更新動作である。リストがオブジェクト・データベース850内の一次キーであり、リストに関連するキーワードが（リストの名前ではなく）このキーを介してリストに関係付けられるので、リストの名前を修正しても、リストに関連するキーワードが影響を受けることはない。言い換えれば、キーワードは、名前を変更した後もリストにリンクされたままである。

好ましい実施形態のR e n a m e 関数を以下に示す。

```
BOOL WINAPI EXPORT RenameList(const char FAR* sOldList,
                                const char FAR* sNewList);

BOOL WINAPI EXPORT RenameProject(const char FAR* sOldProject,
                                  const char FAR* sNewProject);

BOOL WINAPI EXPORT RenameContact(const char FAR* sOldContact,
                                  const char FAR* sNewContact);

BOOL WINAPI EXPORT RenameKeyword(const char FAR* sOldKeyword,
                                  const char FAR* sNewKeyword);
```

上記の関数のいずれかの関数呼出しを1回行うだけで名前変更プロセス全体が処理され、クライアントのタスクが簡略化される。

パーサからの結果の検索

パーサ300から情報を検索するには、クライアントはバッファを割り付け、関数呼出しを介してパーサ300 D L Lに渡さなければならない。解析情報は、以下のように定義される出力データ構造解析830を使用して検索される。

```
typedef struct_ParseOutput{
    char sList[BUFFER_SIZE];
    char sProject[BUFFER_SIZE];
    char sContact[BUFFER_SIZE];
    char sDelegate[BUFFER_SIZE];
    char sDate[BUFFER_SIZE];
    char sTimeBegin[BUFFER_SIZE];
    char sTimeEnd[BUFFER_SIZE];
    char sDateEnd[BUFFER_SIZE];
    long nFrequency;
    BYTE bRecurring;
    BYTE nUnits;
    BYTE nWeekdays;
    BYTE nWeekdayPos;
}ParseOutput, *ParseOutputPtr;
```

パーサ300 DLLのクライアントは、解析出力データ構造 (Visual Basicの「struct」または「type」) を割り付け、以下の関数呼出しのどちらかを使用して解析すべき入出力表現と共に渡す。

Parse()

DateParse()

キーワードの提案は、以下のように定義されるキーワード提案構造を使用してパーサ300から検索される。

```
typedef struct_KeywordSuggestion{
    char sKeyword1[BUFFER_SIZE];
    char sKeyword2[BUFFER_SIZE];

    char sKeyword3[BUFFER_SIZE];
}Keyword Suggestion, *KeywordSuggestionPtr;
```

解析DLLのクライアントは、キーワード提案データ構造を割り付け、以下の関数呼出しのどちらかを使用して解析すべき入出力表現と共に渡す。

```
SuggestListKeywords()
SuggestProjectKeywords()
SuggestContactKeywords()
```

自動完了（自動充填）は文字バッファしか必要とせず、このバッファは、CまたはC++では

```
char sBuffer[BUFFER_SIZE];
```

として、あるいはVisual Basicでは

```
Dim sBuffer As string *BUFFER_SIZE
```

としてクライアントによって宣言することができる。この場合、このバッファは、完成すべき表現と共に、以下の関数呼出しのどちらかに渡される。

```
AutoFillList()
AutoFillProject()
AutoFillContact()
```

再発日付の処理

本発明は、解析出力データ構造の以下のフィールドを使用することによって再発日付解析をサポートする。このフィールドを以下に示す。

nFrequency: Long. スル値=0。「5日」→nFrequency=5。

nUnits: BYTE. スル値=0。日=1、週=2、月=3、年=4

nWeekdays: BYTE. スル値=0。日曜日=1、月曜日=2、火曜日=4、水曜日=8、木曜日=16、金曜日=32、土曜日=64。曜日のすべての可能な組合せを記憶することができる。たとえば、土曜日および日曜日=64+1=65である。月曜日、火曜日、および水曜日=14である。nWeekdaysが特定の曜日を含むかどうかを判定するには、その曜日にnWeekdaysでビット積演算を施す（C/C++では&演算子、Visual Basicでは「and」演算子）。ビット演算の結果がゼロである場合、その曜日は

nWeekdaysには含まれず、そうでない場合は含まれる。

nWeekdayPos: BYTE.ヌル値=0。所与の月の第1曜日、第2曜日、第3曜日、第4曜日、または第5曜日。したがって、「7月の第2月曜日」はnWeekdayPos=2を意味する。

最後に、単純日付を処理しているのか、それとも再発日付を処理しているのかを示すブール値が与えられる。

bRecurring: BYTE. 擬=0。真=0以外。bRecurring=真である場合、sDateは再発イベントの開始日付として働く。

パーサ300は終了日付を必要としない。すなわち、再発イベントは「永遠に」継続するものと仮定される（ユーザは、完全な範囲を指定する場合はユーザ・インタフェース200を使用しなければならない）。それにもかかわらず、指定された終了日付をサポートするために、解析出力データ構造内に「sEndDate」フィールドが設けられる。しかし、再発日付を解析するときには常に開始日付が与えられる。パーサ300は、開始日付を必要としないが（すなわち、1997年8月8日に始まる毎金曜日）、再発イベントについては常に開始日付を

（インテリジェントに）推量することを試みる。この開始日付は、sDateフィールド中の解析出力構造を介して解析される。

コラボレーション

本発明の好ましい実施形態ではコラボレーション・サポートが実現される。パーサ300は各キーノートの開始位置で2つの可能性を探す。他の同様なキーワード・トリガリング・コラボレーションも実現できることに留意されたい。

- ・語「please」（または「pls」）と、そのすぐ後に続く担当者キーワード

- ・担当者キーワードと、そのすぐ後に続く語「please」（または「pls」）

パーサ300は、担当者キーワードと語「please」の間の単一または複数の句読点（コンマ、ピリオド、コロンの、およびセミコロン）も受け入れる（たとえば、「Brian. Please...」または「Please, Brian

n. . .」)。

パーサ300は、このキーノート・シーケンスを認識すると、解析出力データ構造のsDelegatフィールドに(担当者キーワードの指し示す)担当者名を充填する。sContactフィールドに次の担当者フィールド(もしあれば)が表示される。たとえば、「Brian Smith」(キーワード=「brian」)および「Danny Jones」(キーワード=「danny」)の2人の担当者について考える。入力表現

「Brian, please call Danny and arrange for all of us to go to lunch tomorrow at 2pm (Brianへ。Dannyに電話して、私達全員が明日午後2時に昼食にでかけるよう手配してください)」

によって、以下の解析出力データ構造フィールドが得られる。

sDate=<tomorrow's date>

sTimeBegin=14:0

sDelegate="Brian Smiga"

sContact="Danny Rabbani"

本発明のコラボレーション・サポートについて以下に詳しく説明する。

オブジェクト・データベース

本発明のオブジェクト・データベース850は、データベース内の1つのタイプのオブジェクトと任意のタイプの1つまたは複数の他のオブジェクトとの任意のアソシエーションをサポートする。また、オブジェクト・データベース850は、2人のユーザが共通のサーバを共有しているか否かを問わず、本発明の2人のユーザ間のコラボレーション(遂行のための折衝および追跡アクション要求を含む)もサポートする。

第15図から第18図を参照すると、オブジェクト・データベース850内で維持される様々なテーブルが示されている。テーブル・キーは例示のために示すに過ぎないことに留意されたい。

オブジェクトのタイプ

第15図に、好ましい実施形態のオブジェクト・タイプ・テーブルを示す。本発明のデータベースで現在サポートされているオブジェクトのタイプはいくつかある。第15図には、これらのタイプのオブジェクトの代表的な一部が、好ましい実施形態における特定のオブジェクトのタイプの説明と共に示されている。オブジェクト・タイプは、オブジェクト・データベース850内のテーブルに格納されるため、実行時にオブジェクト・タイプの追加、削除または修正を行うことができる。これは、ユーザ定義タイプの情報またはオブジェクト（たとえば「GPS位置」）をサポートするのに有用である。

オブジェクト・タイプとその常駐場所のテーブルとのアソシエーション

第15図にリストされているすべてのオブジェクト・タイプは、オブジェクト・データベース850内のいくつかのテーブルのうちの1つのテーブルに格納される。オブジェクト・タイプとその常駐場所のテーブルとの間のアソシエーションは、第16図に示すオブジェクト・アソシエーション・テーブルで保持される。第16図のオブジェクト・アソシエーション・テーブルに示すように、同じテーブルに複数のオブジェクト・タイプを関連づけることができ、したがって同じテーブルに実際に格納することができる。たとえば、Person（個人）、Delegate（被委任者）、FYI（参考情報）、およびAttached Person（付加個人）というオブジェクト・タイプはすべて、「People（個人群）」テーブルに関連づけられる。

オブジェクト・リンク・テーブル

好ましい実施形態のリンク・テーブルは、本発明のデータベースに入っており、1つのオブジェクトと任意のタイプの別のオブジェクトとの自由なアソシエーションを可能にする特別なテーブルである。たとえば、リンク・テーブルによって、本発明は「個人」オブジェクト・タイプをEメール・アドレス・オブジェクト・タイプに関連づけることができる。同じ「個人」を、それぞれリンク・テーブルで別々の項目として表されている追加のEメール・アドレスとも関連づけることができることに留意されたい。

第18図を参照すると、サンプル・リンク・テーブルが図示されている。図の

ように、リンク・テーブルの欄（構造）には、キー 1、タイプ 1、キー 2、およびタイプ 2 という、2 つのキーと 2 つのオブジェクト・タイプの指定が含まれている。

本発明のオブジェクト・データベース 850 内のすべてのオブジェクトは、そのオブジェクトに関連づけられた固有の識別子、またはキーを有する。これらのキーは、特定のテーブル内のオブジェクトを記述するレコードまたは項目の一部として格納される。たとえば、ブライアン・スミガは 101 というキーを持つ「

Person（個人）」オブジェクト・タイプのインスタンスである。ブライアンの名および姓と、彼のオブジェクト・インスタンス・キーは、単一のレコードの一部として「People（個人群）」テーブルに直接格納されることになる。

前述のように、本発明のデータベース内のすべてのオブジェクトはそのオブジェクトに関連づけられたタイプも有する。1 つのオブジェクトのオブジェクト・キーおよびオブジェクト・タイプと、別のオブジェクトのキーおよびタイプが与えられれば、第 18 図にサンプルを示すリンク・テーブル内の単一の項目（レコード）を介してその 2 つのオブジェクトを「リンク」することができる。

たとえば、「ブライアン・スミガ」は、キー 101 を有する「Person（個人）」オブジェクト・タイプのインスタンスを表し、「smiga@actioneer.com」がキー 102 を有する対応する「Email Address（Eメール・アドレス）」のインスタンスを表す場合、「ブライアン・スミガ」オブジェクト・インスタンスを以下のようにリンク・テーブル内の対応する「smiga@actioneer.com」Eメール・アドレス・インスタンスにリンクすることができる。

キー 1	タイプ 1	キー 2	タイプ 2
101	5	102	13

上記で、「Person（個人）」オブジェクト・タイプはキー 5 を有し、「Email（Eメール）」オブジェクト・タイプはキー 13 を有する。上記のリンク・テーブル項目の例は、タイプ「Person（個人）」（5）のブライアン

・スミガ (キー=101) がタイプ「Email Address (Eメール・アドレス)」(13) の smiga@actioneer.com (102) に関連づけられる (すなわちリンクされる) ことを示している。好ましい実施形態では、項目は常にタイプ1の値がタイプ2の値以下になるようにリンク・テーブルに格納される。このテーブル構成によって探索が容易になる。

サンプル・データベース表現

次に第17図を参照すると、本発明のオブジェクト・データベース850内の様々なテーブルの構成と使用法を示す例が図示されている。各テーブルに図示されている欄は、本発明のデータベースに実際に入っている欄のサブセットに過ぎない。たとえば、「Project (プロジェクト)」は関連づけられた「Outcom (結果 (目標))」も有し、これは「プロジェクト」テーブル内の別個の欄に保管されることになる。プロジェクトに関する追加情報は、「Simple Date (単純日付)」(プロジェクトの開始日、期限日、および完了日を含む) などの別のテーブルに入れられ、前述のようにリンク・テーブルを介して関連づけられた「Project (プロジェクト)」にリンクされる。「Contact (担当者)」テーブル内の担当者のいずれかに関する追加情報は、「物理アドレス」テーブル、「Eメール・アドレス」テーブルなどに入れられ、前述のようにリンク・テーブルを介して関連づけられた「Person (個人)」にリンクされる。第17図および第18図に示すテーブル表現で、リンクされた項目の記述を示すイタリック体の欄は、好ましい実施形態のデータベースには実際には現れないことに留意されたい。これらは、本特許出願における説明のために示すものである。

第17図に示す初期データベース・テーブルの内容が与えられたとして、「デニス・バックハイム」という名前のユーザを考え、キーノート領域220にテキスト表現を入力することによって、以下のようなサンプル・キーノートを作成するものとする。

「Brian, please call Jim tomorrow re
patent status (ブライアン、どうぞ明日ジムに特許の状況に関

して電話してください) 」

さらに、ユーザは「ジム・ソルター」という「個人」と「特許」という「プロジェクト」を付加し、上記で詳述したようにトム・ハーガンに参考情報(FYI) キーノートも送るものとする。この例では、本発明のパーサ300が、第17図に示すテーブルで確立する必要があるリンクの多くを自動的に判断する。すなわち、「ブライアン」というキーワードは、このキーノートの「Delegat

e (被委任者)」である「ブライアン・スミガ」にリンクされたキーワードとして認識される。「call (電話)」というキーワードは、「Calls (電話)」リストにリンクされたキーワードとして認識される。「ジム」というキーワードは、「ジム・ソルター」という担当者個人にリンクされたキーワードとして認識される。「tomorrow (明日)」は、97年2月8日(または現在日付に1日を足した日付)を意味するものとして認識される。「patent (特許)」とは「Patents (特許)」プロジェクトにリンクされたキーワードとして認識される。残りのリンクは、この新しいキーノートが作成されたユーザ・インタフェース200によって決定される。

ユーザであるデニスによってこの新規キーノートがファイルされると、本発明によって以下の重要なデータがオブジェクト・データベース850に記録される。

- ・ 被委任者に送られるキーノートの入れ物としてエンベロープ(タイプ=0)が付加される。エンベロープは、ノートが使用されたときまたは標準Eメール・クライアントで受け取られるときのノートの主題などのデータを含む。このエンベロープには、オブジェクト・データベース850によって212というデータベース・キーが割り当てられているものとする。

- ・ ノートのためにリスト項目(タイプ=1)が追加され、「Brian, please call Jim tomorrow re patent status (ブライアン、どうぞ明日ジムに特許の状況に関して電話してください)」と書かれていることになる。このリスト項目にはオブジェクト・データベース850によって213というキーが割り当てられているものとする。

・ この例ではこの新規キーノートのためにいくつかのリンク・テーブル項目が作成されることになる。このサンプル入力キーノートの結果として作成されるサンプル・リンク・テーブルを第18図に示す。図のリンク・テーブル内のこれらの項目（および1または2個のより重要でない追加の項目）は、例として入力されたノートを記述するのに十分である。

本明細書に記載の本発明のオブジェクト・データベース850実施態様は、本明細書に記載の自然言語パーサ300と共にのみ使用されるものではなく、より広く適用可能であることが、当業者には明らかであろう。他の実施形態では、入力テキスト・データは、構造化レコードまたはバッファとして与えることができ、そこからオブジェクト・データベース850が、第18図に例示するリンク・テーブルを作成するのに必要な情報を取り出す。

本システムの2人以上のユーザ間のコラボレーション

本システムによって、ユーザは自分のアクション／プロジェクトと時間をより効果的に管理することができる。多くの場合、アクション／プロジェクトは2人以上で取り扱わなければならない。後述するコラボレーション・サイクルによって、アクション／プロジェクトのさらなる遂行のためのユーザ間の対話が可能になり、本システムのユーザ間で情報を効率的に交換することができる。

前述の第1図に、本発明の一実施形態が実施される典型的なデータ処理システムが図示されている。本発明は、アクション要求の発信者のために本システムの少なくとも1つのインスタンスを使用することを理解されたい。一実施形態では、前述の通信装置125によって、ユーザは後述するように協働することができる。あるいは、ユーザのコンピュータ・システムを直接結合することもできる。あるいは、複数のユーザが同じコンピュータ・システムを使用することもできる。

入力テキストを構文解析するほかに、前述のパーサ300はさらにキーワードを使用して、入力されたテキストを「分類」する。一実施形態では、メッセージ・タイプには共用メッセージと個人メッセージがある。共用メッセージとしては、FYI（参考情報）アクション要求がある。個人メッセージとしては、個人ア

クションと個人メモがある。当然ながら、その他のメッセージ・タイプも使用可能である。

アクション要求とは、少なくとも1人の他の個人に対して送出され、応答を必要とする入力テキストである。アクション要求は一般に、他のユーザに何かを行

うように依頼するものである。一実施形態では、「please (どうぞ～して下さい)」というキーワードによってアクション要求が開始される。前述のように、ユーザの裁量でリストに他のキーワードも追加することができる。たとえば、ユーザはキーワード「I need you to (あなたに～してもらいたい)」、「pls (どうぞ)」、または入力テキストがアクション要求であることを示す同様の語を付加することができる。したがって、「Joe, I need you to call Bill about the Project X deadline (ジョー、ビルにプロジェクトXの期限について電話してもらいたい)」という文は、ジョーに宛てられたアクション要求として分類される。担当者はビルであり、プロジェクトは「プロジェクトX」である。要求者は日付を含めなかったため、日付はない。このような解析は、前述の構文解析方法を使用して行うことができる。しかし、本発明のシステムは前述の構文解析方法には限定されない。宛先、担当者、プロジェクト、および日付を識別する他の方法も使用可能である。

FYI (参考情報) または「for your information (参考のために)」は、少なくとも1人の他の個人に送出され、応答を必要としない入力テキストである。これは、他の人に情報を知らせるために使用される。たとえば、FYIを使用して、新しいマネージャが加えられたことを他の人に通知することができる。FYIタイプのためのキーワードは、「FYI」、「For your information」、「Please note」、およびユーザがキーワード・リストに含めた他の任意のキーワードとすることができる。一般に、FYIの宛先は、このキーワードの近くから導き出される。たとえば、「John, FYI, Tom is in charge of Project X now. (ジョン、FYI、現在トムがプロジェクトXを担当していま

す。) 」というFYIの場合、このFYIのコピーがジョンに送られることになる。

個人アクション／メモは、誰にも送られないテキストである。他のキーワードがないことによって、項目が個人アクション／メモとして分類されることが示される。たとえば、「Call Jim about Project X (ジム

にプロジェクトXに関して電話する)」という入力テキストは個人アクションである。これは、前述のようにユーザ自身のシステムにファイルされ、他の誰にも転送されない。

コラボレーションの枠組み

本発明のシステムは、いくつかの異なる枠組み上で実施することができる。第19図は、クライアントーサーバ・システムの図である。このクライアントーサーバ・システムは、サーバ1910と、サーバ1910に接続された複数のクライアント1920とから成る。サーバ1910は、ユーザとデータベースの間で送信されるアクション要求に関する情報を記憶して対話をサポートする。これは、クライアントーサーバ・システムにおけるサーバの存在を前提とし、サーバを必要とする。これによって、コストがかかり、複雑になるだけでなく、サーバ2010へのアクセスを共用する限られたグループのメンバーのみにコラボレーションが限定される。言い換えると、サーバ1910へのアクセス権を有するクライアントC1... Cn2020の閉ループに限定される。これは、コラボレーション・システムをクライアントーサーバ枠組みの外部の個人との対話に使用することができないことを意味する。

第20図は、ピア・ツー・ピア分散システムの図である。複数のピア（対等）端末P1... Pn2030が相互接続されている。ピア・ツー・ピア・システムでは、任意の2つのピア端末が、直接リンクされていなくてもシステムを介して通信することができる。この分散システム・モデルはインターネットに似ている。したがって、任意の2人の個人が、両方ともEメール・アドレスにアクセスすることができる限り、ピア・ツー・ピア分散システムを使用して通信することができる。一実施形態では、ピア・ツー・ピア分散システムを拡張して、ボイス

・メール、携帯情報端末、およびメッセージの送受信が可能なその他の任意の機構などの機構を組み込むことができる。

分散ピア・ツー・ピア・システムは、本発明のシステムを使用しないユーザとの通信を可能にする。したがって、本発明のシステムによって、受信者が本発明のシステムの加入者であるか否かを問わず、すべてのアクション要求および予定

表をシームレスに統合することができる。一実施形態では、このピア・ツー・ピア分散システム・モデルが本発明に使用される。

アクション要求の追跡

分散システムの使用には、アクション要求を追跡する代替手段を必要とする。各アクション要求を追跡するサーバがないため、各アクション要求と、プロジェクト、送信者などのそれに付随するオブジェクトとを識別する方法が必要である。この方法によって、システムは応答を適切なアクションと一致させることができる。

一実施形態では、外部キー・テーブルを使用してアクション要求を追跡する。わかりやすいように、第15図から第18図に関して前述した例を続けて使用する。当該アクション要求は以下の通りであった。

被委任者：ブライアン 発信者： デニス FYI： トム 件名： アクション要求：Please Brian, call Jim ... エンクロージャ： プロジェクト情報、担当者情報
テキスト：Please Brian, call Jim re: patent project. (ブライアン、特許プロジェクトに関してジムに電話して下さい。)

上記のアクション要求の形式は、本発明のシステム上で表示されるアクション

要求の実際の形式を反映したものではない。このようなアクション要求の概観については後述する。このアクション要求は、デニスによって入力され、ブライアンに送られた。さらに、このアクション要求のFYIコピーがトムに送られた。これは、トムにこのアクション要求に対する注意を促すためのものである。アク

ション要求が生成されると、アクション要求を入れるエンベロープが生成される。エンベロープには、件名、リスト項目へのリンクと、受信者のEメール・アドレスが含まれる。以下では、ブライアンのシステムで行われるアクションについてのみ説明する。しかし、同様の活動はトムのシステムでも行われる。

ブライアンがアクション要求を受け取ると、この新規アクション要求のためにいくつかのリンク・テーブル項目が作成される。これらのリンク・テーブル項目は、前述のデニスのテーブルの項目と似ている。ローカル・キー番号は個人のデータベースに固有のものであるため、それらのキー番号は異なることがある。さらに、ブライアンのシステムは、外部キー・テーブルに入れられるいくつかの項目を生成する。外部キー・テーブルは、送信者のシステムの一部である1つのデータベース内のコラボレーションアクション要求を、受信者のシステムの一部である別のデータベース内の同じアクション要求にマッピングする際に使用される。このリンクは、対応するリスト項目が入ったエンベロープを関連づけることによって示される。

作成者	ローカル・データベース	ローカル・キー	外部データベース	外部キー
S M T P	ブライアンのデータベース I D	ブライアンの受信エンベロープ・キー	デニスのデータベース I D	デニスの送信エンベロープ・キー (212)
S M T P	ブライアンのデータベース I D	ノート作成者 (デニス) のためのブライアンのキー	デニスのデータベース I D	デニスのための「me」キー
S M T P	ブライアンのデータベース I D	作成者の E メール・アドレス (デニス) のためのブライアンのキー	デニスのデータベース I D	デニスのための「my e-mail」キー
S M T P	ブライアンのデータベース I D	ブライアンのための「me」キー	デニスのデータベース I D	被委任者 (ブライアン) のためのデニスのキー
S M T P	ブライアンの	ブライアンのた	デニスのデー	被委任者の E メ

	データベース ID	めの「my e -mail」キ ー	データベースID	ール・アドレス (ブライアン) のためのデニス のキー
S M T P	ブライアンの データベース ID	リンクされた個 人(ジム)のた めのブライア ンのキー	デニスのデー タベースID	リンクされた個 人(ジム)のた めのデニスのキ ー
S M T P	ブライアンの データベース ID	F Y I 受信者の Eメール・アド レス(トム)の ためのブライ アンのキー	デニスのデー タベースID	F Y I 受信者の Eメール・アド レス(トム)の ためのデニス のキー
S M T P	ブライアンの データベース ID	リンクされたプ ロジェクト(特 許)のためのブ ライアンのキー	デニスのデー タベースID	リンクされたプ ロジェクト(特 許)のためのデ ニスのキー
S M T P	ブライアンの データベース ID	送信者のための ブライアンのキ ー	デニスのデー タベースID	送信者のための デニスのキー
S M T P	ブライアンの データベース ID	送信者Eメール のためのブライ アンのキー	デニスのデー タベースID	送信者Eメール のためのデニス のキー

表 1

作成者の欄は、外部キー・テーブルの項目の作成者である。作成者欄には、アクション要求を受け取った装置が表示される。この例では、これは、デニスからのアクション要求を受け取った簡易メール転送プロトコル(SMTP)プラグインだった。

ローカル・データベース欄には、被委任者(ブライアン)のデータベースを識別する固有識別子が入る。外部データベース欄には、要求者(デニス)のデータベースを識別する固有識別子が入る。一実施形態では、識別子はグローバル・ユニーク・アイデンティファイア(GUID)であり、GUIDが生成される端末のイーサネット・アドレスとGUIDが作成された時刻とに基づく128ビット

値である。あるいは、その他の固有識別子も使用することができる。

ローカル・キーと外部キーは、個人のデータベース・テーブルに入っているキー番号である。第X図に関して前述した例示のキーは、外部キー・テーブル、すなわちデニスのデータベース項目に含まれている。たとえば、ノート作成者（デニス）のためのローカル・キーは、ブライアンのデータベース内のデニスに関連づけられたキー番号である。「me」キーは特殊目的値であり、これは、ユーザのデータベースにユーザ自身を識別する情報が入っているかどうか確実ではなく、ユーザ名をテキストの比較によって突き合わせる方法では信頼性がないために使用される。この方法によって、一致を示すことができる。「my e-mail」キーは、データベース所有者のEメール・アドレスのための類似キーである。さらに、デニスがブライアンから応答を受け取るとデニスのシステムで類似のテーブルが生成される。当然ながら、デニスのシステムでは、デニスのデータベースIDはローカル・データベースであり、ブライアンのものは外部データベースである。

アクション要求にリンクされた「個人」および「プロジェクト」と、アクション要求の作成者（別の「個人」）、アクション要求の送信者（別の「個人」）、被委任者、およびアクション要求に関連づけられたEメール・アドレスについての外部キー項目も、両方の側（要求者と被委任者）で作成される。「個人」、「プロジェクト」などの項目は、応答時に正しいリンクが作成／維持されるようにするためと、いずれのユーザのデータベースでも重複項目が作成されないように保証するために作成される。

代替実施形態では、送信者と受信者のEメール・アドレスは、アクション要求自体に付加された固有識別情報と結びついて、アクション要求を識別する。他の代替実施形態では、送信者／受信者の公開鍵／私用鍵と結びついた識別情報固有ローカルを使用して各アクション要求を識別する。

コラボレーション・サイクル

第21図は、本発明の概要を示すフローチャートである。ブロック2100で、本発明のコラボレーション・サイクルが開始される。これは、要求者が1人ま

たは複数の被委任者にメッセージを送信することによって開始される。

ブロック2110では、要求者のシステムと1人または複数の被委任者のシステムとの間でのコラボレーションがある。このコラボレーションには、最終回答に達するように設定された複雑な一連の折衝ステップを必要とする。これについては後で詳述する。

ブロック2120で、プロセスは最終回答が肯定か否定かまたは完了かを問う。両当事者がタスクを行うことに合意した場合は肯定回答になる。両当事者がタスクを行わないことに決めた場合は否定回答になる。

最終回答が肯定の場合、プロセスはブロック2130に進む。ブロック2130で、最終回答が適切なカレンダーおよびリストにファイルされる。これには、該当する場合には、担当者、プロジェクトに関連づけられたリストが含まれる。以下では、これらのリストをプロジェクト/担当者リストと呼ぶ。後述するように、ファイル先は元のアクション要求の解釈と、コラボレーションプロセスとに応じて決まる。

最終回答が否定の場合、プロセスはブロック2140に進む。ブロック2140で、アクション要求と、元のアクション要求のコラボレーション更新が、カレンダーとリストから削除される。このプロセスについては後で詳述する。

ある時点で、否定応答を受け取らなかった場合、被委任者は要求者に「完了」応答を送る。「完了」応答は被委任者が後述するプロセスを使用して送信することができる。あるいは、被委任者が自分のカレンダーおよび/またはプロジェクト/担当者リストでそのアクション/プロジェクトに完了済みの印を付けると、自動「完了」応答が生成され、要求者に送られる。

要求者が「完了」応答を受け取ると、要求者のシステムで元のアクション要求に完了のマークが付けられる。さらに、一実施形態では、自動肯定応答書式が生成される。一実施形態では、肯定応答書式は、「Thank you for completing my action request regarding the (プロジェクト名) project. (「プロジェクト名」プロジェクトに関するアクション要求の完了ありがとうございます)」のような汎用テキストから成る。一実施形態では、異なる被委任者に異なるタイプの肯定

応答を関連づけることができる。たとえば、要求者は、お礼ボイス・メールまたは文書を受け取るべき特定の被委任者を識別する。その場合、システムがそのような肯定応答を自動的に生成することができる。一実施形態では、ユーザは肯定応答のタイプを選択することができる。一実施形態では、要求者は肯定応答を送るべきであるという通知を受け取るだけである。

第22A図は、発信者、または要求者から見たコラボレーション・サイクルを示すフローチャートである。プロセスは、ブロック2200で、入力テキストが構文解析され、アクション要求として分類されると開始される。一実施形態では、これはパーサが「please」などのキーワードを検出すると行われる。

ブロック2205で、プロセスは被委任者を識別する。アクション要求は1人または複数の当事者に宛てることができる。これらの当事者は被委任者である。前述のように、一実施形態では、キーワード「please」の前または後の名前が被委任者とみなされる。前述のように、ユーザは、当該項目がアクション要求であることを示す追加のキーワードを付加していることがある。

被委任者は、個人またはグループとすることができる。たとえば、アクション要求は「Managers（マネージャ各位）」に宛てることができる。その場合、「Manager（マネージャ）」というキーワードには複数のマネージャが含まれることがある。一実施形態では、複数被委任者の場合、各被委任者ごとに別々のアクション要求が生成され、各被委任者は個別に扱われる。説明を簡単にするために、このフローチャートのこれ以降の部分では、単一の被委任者しかいないものとする。

一実施形態では、被委任者が識別されるとヘッダが生成される。一実施形態では、このヘッダはいくつかのフィールドを含む。一実施形態では、これらのフィールドには、「Delegate（被委任者）」、「FYI」、および「Enclosures（エンクロージャ）」が含まれる。「被委任者」フィールドには、前述のようにして判断された被委任者が含まれる。FYIフィールドも同様の方法で判断される。一実施形態では、「被委任者」フィールドは他のメッセージ交換アプリケーションの「To（宛先）」フィールドにマップされ、FYIはCCフィールドにマップされる。

「Enclosures (エンクロージャ)」フィールドによって、要求者はアクション要求に様々な項目を付加することができる。一実施形態では、エンクロージャには、アクション要求に関連するプロジェクトおよび担当者に関する任意のファイルまたは情報を含めることができる。これが特に有用なのは、本システムを使用しない被委任者にアクション要求を送る場合である。その場合、被委任者が受け取ったメッセージは、担当者またはプロジェクトに関する情報を持つデータベースにリンクすることができない。したがって、これらの項目を付加することによって、要求者は受信者の作業を簡略化することができる。

「From (発信元) :」フィールドに要求者の名前が入れられる。ヘッダには件名も含めることができる。件名は、アクション要求または項目が解析された結果のプロジェクトの最初の数語とすることができ、アクション要求の作成者が入力することもできる。一実施形態では、件名は「アクション要求：＜要求の最初の数語＞」、または「FYI：＜FYIの最初の数語＞」のようになる。これによって、受信メッセージのタイプが受信者に明らかになるだけでなく、メッセージの主題に関するある程度の情報が提供される。

ブロック2210で、アクション要求が被委任者に送られる。アクション要求は、電子メールまたはその他の任意の手段によって送ることができる。一実施形態では、アクション要求は、本発明のシステムにおける被委任者の受信箱に直接送られる。一実施形態では、被委任者が本システムを使用していない場合、アクション要求は被委任者のアドレスに送られる。一実施形態では、これには被委任者のEメール・アドレス、ファクス番号、ボイスEメール番号またはページが含まれる。

本出願に記載のファイリング・アクションおよび同様のアクションの多くは、本発明のシステムの使用を必要とする。しかし、アクション要求は、テキストまたは音声を何らかの形で受け取ることができる受信オブジェクトを有するどのような個人にでも送信することができる。一実施形態では、被委任者がEメール接続を備えない場合、その被委任者に対してはアクション要求をファクス送信することができる。一実施形態では、本システムはアクション要求をリッチ・テキスト(RTF)形成にフォーマットすることができ、それを被委任者にファクス送

信することができる。一実施形態では、アクション要求をボイス・メール・メッセージに変換し、本システムで送信することができる。他の実施形態では、アクション要求はページャ・メッセージとしてページャに送信することができる。

一実施形態では、アクション要求は、ユーザが読めるASCII形式にフォーマットされる。一実施形態では、アクション要求は以下のように再フォーマットされる。

「元のアクション要求のテキスト」

これは、「要求者」から「被委任者」に対するアクション要求です。

これは下記の事項に関連する。

プロジェクト：「プロジェクト」

担当者： 「担当者」

期限日： 「期限日」

添付書類：「同封物」

引用符内のテキストは、アクション要求からの情報に基づいて挿入される。このテキスト形式は、送信され、本システムの加入者ではない被委任者が読むことができる。

ブロック2215で、アクション要求がファイルされ、適切なカレンダーおよび／またはプロジェクト／担当者リストに鉛筆書きされる。一実施形態では、アクション要求は「待機」リストにファイルされる。これは、まだ解決されていないアクション要求が入られるリストである。一実施形態では、アクション要求のコピーも、解析されたプロジェクト／担当者リストにファイルされる。さらに、該当する場合には、アクション要求は解析された任意のリスト、プロジェクト、担当者、およびカレンダーに鉛筆書きされる。「鉛筆書き」とは、カレンダー

またはプロジェクト／担当者リストに異なる色で記入することを示す。これによって、ユーザはまだ合意していない項目を容易に識別することができる。一実施形態では、黒または青色の通常の項目に対して、鉛筆書きされた項目は灰色で表示される。

ブロック2220で、プロセスは被委任者からの応答を待つ。プロセスは待機

状態のままにはなっておらず、他のプロセスの実行を続ける。しかし、このアクション要求サイクルは完了を待つ。被委任者からの応答を受け取ると、プロセスはブロック2225に進む。被委任者からの応答は要求者の受信箱に現れる。一実施形態では、応答がない場合、期限日の前に催促が送られる。

ブロック2225で、プロセスは応答が肯定、否定、またはその他のカテゴリのいずれであるかを問う。これらのカテゴリについては、第22B図を参照しながら後述する。一実施形態では、被委任者が事前設定された選択項目または応答から選択することによって応答リストが選択される。代替実施形態では、適切な応答リストを判断するために応答が再解析される。一実施形態では、日付が再解析される。代替実施形態では、応答が全部再解析され、再分類される。

応答が肯定の場合、プロセスはブロック2230に進む。ブロック2230で、元のアクション要求が更新され、再ファイルされる。前述のように、アクション要求は最初は「待機」リストに入っている。今度は、アクション要求は解析の結果の判断に従って適切なリストに分類し直される。アクション要求はカレンダーおよびプロジェクト/担当者リストにもインクで入れられる。一実施形態では、インクでの書込みは、鉛筆書きとは異なる色で要求を記入することを含む。一実施形態では、インクの色は黒または青色である。

応答が否定の場合、プロセスはブロック2235に進む。ブロック2235で、アクション要求はカレンダーおよび「Waiting for (待機)」カテゴリから削除される。被委任者は要求の遂行を拒否したため、この時点でプロセスは打ち切られる。しかし、否定応答は被委任者の受信箱に残る。したがって、要求者は、そのプロジェクトを再割当てしたい場合、受信箱内のコピーを使用して再割当てすることができる。

その他の応答の場合、プロセスはブロック2240に進む。ブロック2240で、要求者と被委任者が折衝する。折衝プロセスについては第23図を参照しながら詳述する。折衝が完了すると、プロセスはブロック2245に進む。ブロック2245で、プロセスは再び、最終回答が肯定か否定かを問う。最終回答が肯定の場合、プロセスはブロック2230に戻る。最終回答が否定の場合、プロセ

スはブロック 2235 に戻る。

第 22B 図は、受信者または被委任者から見たコラボレーション・サイクルを示すフローチャートである。ブロック 2250 で、受信アクション要求のコピーが被委任者の受信箱に入れられる。一実施形態では、受信箱は本発明のシステムの一部である。代替実施形態では、受信箱は被委任者の E メール・ボックスとすることができる。一実施形態では、被委任者が本システムのユーザでない場合、到着種類受けは被委任者の E メール・アドレス・ボックスである。一実施形態では、受信箱は、受信したアクション要求が入れられる、本発明のシステムにおけるリストでもある。

ブロック 2250 で、開かれたアクション要求が被委任者に表示される。一実施形態では、被委任者が本発明のシステムを使用している場合、アクション要求は第 4 図に関して前述した形式で表示される。被委任者が本発明のシステムを使用していない場合、一実施形態では被委任者はアクション要求を E メール・メッセージとして開くことができる。被委任者が本発明を E メールとして開く場合、人間が読むことができる単純なテキストまたはフォーマットされたテキストとして表示される。テキストの形式は、前掲の通りである。代替実施形態では、被委任者がリンク可能ないずれかの形式の情報を開く場合、同封物がそのテキストにリンクされる。一実施形態では、被委任者がアクション要求をウェブ・ブラウザで開く場合、ハイパーテキスト・マークアップ言語 (HTML) 形式で表示される。一実施形態では、同封物が文書の最下部に表示される。一実施形態では、HTML では、同封物はテキスト内の該当する情報にリンクすることができる。たとえば、要求者が担当者情報を同封した場合、担当者名はその同封物にリンクされる。したがって、被委任者がその担当者名を選択すると、同封された担当者情報が表示される。テキストを表示するその他の手段も当技術分野で周知である。

アクション要求が開かれると、被委任者はそのアクション要求をさらに委任することができる。当然ながら、被委任者は、本システムを使用しない場合、標準 E メール通信を介さなければこれを行うことができない。この時点で、一実施形態では、ユーザはアクション要求に「Please' new delegate

」を加えることによってアクション要求をさらに他の人に委任することができる。これによって、再解析の際に、アクション要求がその新しい被委任者に宛てて送られる。代替実施形態では、ユーザはヘッダで新規被委任者を手動選択することができ、それによってアクション要求を転送することができる。アクション要求は、その新規被委任者に宛て直されて転送される。この時点で元の被委任者は要求者になる。しかし、元の被委任者は依然として元の要求者に対して応答しなければならない。

ブロック2260で、ユーザは応答を入力するように促される。一実施形態では、ユーザが受信箱内のアクション要求を開くと、表示アクション要求上の応答ボックス内の応答分類選択項目と共に表示される。一実施形態では、応答ボックスはプルダウン・メニューである。代替実施形態では、応答ボックスは、ラジオ・ボタン、またはいくつかのリスト項目のうちの1つの選択項目を示すその他の手段を含む。

ユーザはこれらの選択項目の1つをアクション要求に対する応答として選択することができる。一実施形態では、この選択項目には、「Yes (受諾)」、「Yes if (条件付き受諾)」、「No (拒否)」、「Comment (コメント)」、および「Done (完了済み)」が含まれる。「Yes」応答は、被委任者が委任を受け入れ、アクション要求を行う意志があることを示す。「Yes if」応答は、被委任者がアクション要求を実行する意志があるが、反対提案があることを示す。反対提案とは、たとえば会議日を変更する提案などである。「No」応答は、被委任者がその委任を辞退することを示す。「Done」応答は、被委任者が委任を受け入れ、割り当てられたプロジェクトをすでに完了していることを示す。最後に、「Comment」応答は、これらのカテゴリに入らない事柄を示す。たとえば、被委任者が、要求者がプロジェクトを誤解していると感じた場合、この応答を使用することができる。その他の応答選択項目も、本発明の基本的な目的を変更することなく組み込むことができる。一実施形態で

は、これらの選択項目は、被委任者が自分の受信箱に入っているアクション要求を開くと表示される。一実施形態では、ユーザはこれらの選択項目の1つを選択

しなければならない。代替実施形態では、このような応答選択肢は表示されない。その実施態様では、ユーザは自由形式テキストで応答する。その実施形態では、パーサを使用してユーザの応答を解析し、それを上述のカテゴリのうちの1つに入れる。

ユーザがこれらの選択項目の1つを選択すると、応答書式が作成される。応答がYes、No、またはDoneのいずれかの場合、応答書式は完成している。ユーザはそれ以上情報を入力しなくてもよい。しかし、ユーザは他の情報を入力することもできる。一実施形態では、応答がYes、No、またはDoneの場合、応答にヘッダが自動的に付加され、自動的に送信される。ユーザは入力を促されない。

他方、選択された選択項目がYes IfまたはCommentの場合、応答書式が適切なヘッダ情報と共に自動的に生成される。一実施形態では、応答がYes ifだった場合、「Yes, I will do it, If (了承しました。それを実行します。ただし)」などの句が表示され、その後にカーソルが続く。その後で、ユーザはこの句を完成させることができる。代替実施形態ではユーザはこの句を削除して自分の言葉を入力することができる。一実施形態では、Commentボタンが選択された場合、「I have a comment (コメントがあります)」のような句の後にカーソルが表示される。この場合も、ユーザはこの句を完成させるか、消去して別の言葉を書き込むことができる。このようにして、選択された応答選択項目に基づいて応答書式が自動的に記入される。

ブロック2265で、アクション要求と応答が被委任者の適切なプロジェクト／担当者リストおよびカレンダーに入れられる。リストおよびカレンダーに入れられる情報は、解析されたアクション要求および解析された応答に基づく。一実施形態では、日付情報のみが再解析され、他のすべての情報はそのまま残る。被委任者がアクション要求を開くと、仮のプロジェクト／担当者リスト、担当者、およびその他の情報が括弧に入れられてアクション要求のシャドウに示される。

しかし、被委任者はこの情報を、シャドウの中で直接変更するか、または応答内

の情報を変更することによって変更することができる。

ブロック2270で、応答が要求者に送られる。一実施形態では、これはユーザがボタンを押すで行われる。一実施形態では、送信ボタンがある。

ブロック2275で、プロセスは、送ったばかりの応答が肯定、否定、その他のいずれであったかを問う。肯定応答には、「Yes」と「Done」が含まれる。否定応答には「No」が含まれる。「Yes if」および「Comment」応答はその他として分類される。同様に、異なるカテゴリを使用する場合、委任を拒否するカテゴリはすべて否定である。委任をいかなる仕方でも変更しようとして受け入れるカテゴリはすべて肯定である。その他のカテゴリはすべて「その他」である。

応答が肯定の場合、プロセスはブロック2280に進む。ブロック2280で、元のアクション要求が更新され、ファイルされる。前述のように、元のアクション要求は被委任者の「送信箱」リストにある。この時点で、アクション要求は解析結果に基づいて再ファイルされる。アクション要求は適切なカレンダーおよびリンクにもインクで書き込まれる。一実施形態では、要求をカレンダーにインクで入れる操作は、鉛筆書きとは異なる色で要求を記入することを含む。一実施形態ではインクの色は黒または青色である。一実施形態では、応答の通知が要求者の受信箱に残る。他の実施形態では、ユーザは通知を受け取るかどうかを選択することができる。

応答が否定の場合、プロセスはブロック2285に進む。ブロック2285で、アクション要求はカレンダーと「待機」カテゴリから削除される。被委任者は要求の遂行を拒否したため、この時点で要求は打ち切られる。一実施形態では、応答の通知が要求者の受信箱に残る。他の実施形態では、ユーザは通知を受け取るかどうかを選択することができる。

応答が「その他」の場合、プロセスはブロック2290に進む。ブロック2290で、要求者と被委任者が折衝する。この折衝プロセスについては第23図に関して詳述する。折衝が完了するとプロセスはブロック2295に進む。ブロック2295で、プロセスは再び、最終回答が肯定か否定かを問う。最終回答が肯

定の場合、プロセスはブロック2280に戻る。最終回答が否定の場合、プロセスはブロック2285に戻る。

第23図は、折衝プロセスを示す図である。折衝は、前述のようにブロック2240および2290で開始される。最初のメッセージ交換中に要求者と被委任者が合意に達しない場合、両者は折衝プロセスに移る。

ブロック2310で、折衝プロセスが開始される。ブロック2320で、最後のメッセージの受信者が、応答するように促される。被委任者が、「Yes, if」または「Comment」である応答、または肯定でも否定でもないその他のいずれかの応答を返すと、折衝プロセスに入る。

したがって、ブロック2320の最初の反復で、被委任者の初期応答に応答して要求者が応答を促される。要求者の前に表示される応答書式は、被委任者のための応答書式ときわめてよく似ている。

一実施形態では、選択可能なカテゴリは、OK（受諾）、OK If（条件付き受諾）、No（拒否）、Withdraw（撤回）/Done（完了済み）、およびComment（コメント）である。これらのカテゴリは、Yes（受諾）、Yes If（条件付き受諾）、No（拒否）、Done（完了済み）、およびComment（コメント）のカテゴリと類似している。これらのカテゴリ名は、本プロセスに影響を与えることなく変更することができる。OK応答は、被委任者によって提案された変更またはコメントを要求者が受け入れることを意味する。OK If 応答は要求者による反対提案である。Noは被委任者の提案の拒否である。Withdrawは、元のアクション要求を取り消し、被委任者はそのアクション要求に関して何もする必要がなくなったという、被委任者に対する通知である。OKは肯定として分類される。NoまたはWithdrawは否定として分類される。OK IfとCommentは「その他」として分類される。代替実施形態では、応答書式には何もカテゴリが含まれない。その実施形態では、ユーザが自由書式応答を入力する。パーサを使用してこの応答を解析し、その応答が属する応答選択肢を判断する。

ブロック2270および2320に関して前述したように、選択された応答選択肢に基づき、事前フォーマットされた応答が適切なヘッダ情報と共に表示され

る。これによって折衝プロセスが簡略化される。

ブロック2330で、応答が送信され、応答のコピーが適切なカレンダーおよびプロジェクト／担当者リストに入れられる。メッセージも送信された応答に照らしてしかるべく更新される。

ブロック2340で、プロセスは送信された応答が肯定、否定、その他のいずれであったかを検査する。前述の分類で説明したように、被委任者のYes、Doneと要求者のOKは肯定として分類される。被委任者のNoと、要求者のNoまたはWithdrawは否定として分類される。その他のすべての回答、すなわち、Yes If、OK If、およびCommentは、その他として分類される。肯定または否定のいずれかの回答が最終回答である。すなわち、折衝の結論である。回答が肯定または否定だった場合、プロセスはブロック2350に進む。ブロック2350で折衝プロセスは終了する。

ブロック2340で、回答が「その他」であることがわかった場合、プロセスはブロック2360に進む。ブロック2360で、プロセスは起動された自動終了があるかどうかを検査する。一実施形態では、自動終了は要求者が選択することができる選択肢である。自動終了は、事前設定された交換回数後に折衝プロセスを自動的に終了させる。一実施形態では、この折衝終了までの交換回数はユーザが入力する。たとえば、両当事者が5回のEメール交換後に合意することができない場合、プロセスは自動的に終了する。これは、両当事者が合意できないことが明らかな場合に折衝サイクルが果てしなく行われるのを防ぐための方法である。

自動終了プロセスが起動されていない場合、プロセスはブロック2320に戻り、最後のメッセージの受信者に応答を促す。たとえば、最後のメッセージが要求者によって被委任者に対して書かれた場合、プロセスは被委任者にそのメッセージに応答するように促す。

他方、ブロック2360で自動終了プロセスが起動された場合、プロセスはブロック2370に進む。ブロック2370で、最終回答がNoに設定される。これは、要求者と被委任者との間で何の合意にも達しなかったことを示す。プロセスはブロック2350に進み、折衝プロセスは終了する。

第24図は、FYIの配布を示すフローチャートである。前述のように、FYIは発信者から少なくとも1人の受信者に対して送出される。一実施形態では、アクション要求を他のユーザへのFYIとして送信することもできる。FYIは応答を必要としない。一実施形態ではFYIの受信者は回答することができる。

ブロック2410からこのプロセスが開始される。これは、入力されたデータがFYIであることを示すキーワードで始まる情報をユーザが入力すると行われる。FYIは前述のように解析される。ブロック2420で、FYIが発信者のシステムでファイルされる。

ブロック2430で、1人または複数の受信者が識別される。一実施形態では、それがFYIであることを示すキーワードの次に現れる1つまたは複数の名前、またはグループ名が受信者として識別される。ブロック2440で、FYIにヘッダが付加される。ヘッダは「To: (送信先)」フィールドと、「From: (発信元)」フィールドと、「cc:」フィールドと、「enclosures: (同封物)」フィールドと、「subject (件名)」フィールドとを含む。これらのヘッダ項目については、アクション要求に関して前述した通りである。

ブロック2450で、FYIが受信者に送信される。受信者は、FYIを受け取るとそれを開いてファイルすることができる。一実施形態では、配布された応答に対する選択肢は与えられない。このようにすることで、情報に関する無関係の通信が促されない。

第25A図および第25B図は、コラボレーションプロセスの一例を示すフローチャートである。この例では、トムとデニスの2人が特定のプロジェクトに関する会合を設定しようとしている。デニスが要求者、または発信者である。

グラフィック2500で、デニスがテキストを入力している。前述のように、テキストは同時に解析される。したがって、デニスがアクション要求2512に入力していくに従って、プロジェクト、担当者、日付、および適切なプロジェクト/担当者リストがシャドウ2515に現れる。前述のように、解析アルゴリズムは前述のアルゴリズムまたは他の任意のアルゴリズムとすることができる。一実施形態では、デニスがアクション要求2512を入力しているとき、パーサが

そのテキストを解析している。これがアクション要求であることを示すキーワードが見つかったと、アクション要求上にヘッダが入れられる。被委任者、FYI、同封物などが解析アルゴリズムとデニスのアクションによって判断されるにつれて、ヘッダ内にそれらのスペースが追加される。代替実施形態では、アクション要求 2512 は、デニスがテキストの入力が完了したことを示したときにのみ解析される。デニスは、アクション要求の入力を完了すると、その要求を送信する。

グラフィック 2502 では、トムはアクション要求を受け取って開いている。自動的に生成されたヘッダ 2525 が表示されている。その下には、元のアクション要求 2530 が表示されている。ヘッダの下には、応答ブロック 2520 が表示される。応答ブロック 2520 には、可能な応答オプションが表示される。一実施形態では、応答ブロック 2520 はドロップダウン・リストであり、その中から 1 つの応答を選択することができる。他の実施形態では、応答ブロック 2520 は、ユーザが応答を選択することができる他の任意の形式とすることができる。この場合、Yes If 応答がハイライトされている。担当者およびプロジェクト情報については、送信側から導出された提案が最初に受信者に示される。一実施形態では、プロジェクト名と担当者名は括弧に入れられる。これは、プロジェクト名および担当者名がデニスとトムの場合とで同じではない可能性があることを示すためである。前述のように、各ユーザが自分自身のキーワードを入力することができるため、キーワードは様々である。

グラフィック 2504 で、トムは応答ブロック 2520 から Yes If 応答ボタンを選択しており、応答 2545 が表示される。一実施形態では、応答は自動的に「Yes, I'll do it, if... (了承しました。やります。ただし...)」で始まる。この句の後にカーソルが配置され、トムがこの句を完成させることができるようにする。この場合、トムは「we can reschedule for Tuesday. (火曜日に予定変更できれば)」と入力してこのフレーズを完成させている。トムが入力したテキストは、この場合、イタリック体で入れられることによって区別される。実際の応用分野では、テキストをこのようにして区別する必要はないことを理解されたい。シャドウ 2

540内で、プロジェクト名が変更されている。これは、ユーザが手作業で行うことができる。あるいは、プロセスがトムのデータベースとキーワード・リストを使用して元のノートを解析することもできる。この場合、担当者名は同じままである。しかし、プロジェクト名は変更されている。2人のユーザが同じプロジェクトを異なる名前と呼ぶことがあるため、各ユーザが自分自身のプロジェクトに名前を付ける責任を負う。この時点で、トムは応答を送信することができる。その時点で、応答にヘッダが入れられて、要求者、すなわちデニスに返送される。

グラフィック2506で、デニスはトムの応答を受け取り、それを開いている。シャドウ2560に、新しい日付が表示されている。トムは会合のための新しい日付を提案した。この要素のために応答が再解析されたため、シャドウ2560に新しい日付／時刻が表示される。この場合も、応答ボックス2550が表示される。デニスは要求者であるため、多少異なる応答ボックス2550が表示される。項目については、第22B図に関して上述した通りである。この場合は、デニスはYesボタンを選択しており、トムの提案した日付変更に同意している。一実施形態では、デニスがYesボタンを選択した後、応答はシステムによって自動的に送信され、ユーザの関与は終了する。これで最終回答に達した。したがって、折衝は終了する。

グラフィック2508で、デニスのシステムは最終決定情報をファイルする。一実施形態では、最終決定情報は当事者間の通信の履歴を示すボックス内に表示される。元のシャドウ2515は更新されて更新済みシャドウ2570になる。デニスのカレンダーおよびリスト内の元の鉛筆書きされた項目もインクに更新される。これによって、合意に達したことが示される。

グラフィック2510で、トムのシステムは最終決定情報をファイルする。さらに、元のシャドウ2535が、最終合意情報が入った新しいシャドウ2585に更新される。ブライアンのカレンダーおよびリスト内の元の鉛筆書きされた項目もインクに更新される。これによって合意に達したことが示される。

このように、このプロセスによって、本発明の1つの事例についてグラフィックスの形で例示した。実際の画面表示はこの図に示した表示とは同じでないことを

理解されたい。このようにして、要求者と被委任者の間の対話が簡略化される。

これによって、被委任者と要求者はアクション／プロジェクトを遂行するための互いに満足のゆく方法に到達することができる。これは、両当事者に十分な柔軟性を与え、自動化会話タイプの自動化折衝として機能する。

パーサDLLアプリケーション・プログラマーズ・インタフェース (API)

以下の項では、本発明の好ましい実施形態におけるパーサ300DLLアプリケーション・プログラマーズ・インタフェース (API) について詳述する。

ファイル: parseapi.h

目的: 解析300DLL APIを定義する。

注意: このAPI内のすべての関数には、「Prs_」 (Parsの略) というプレフィックスがネームスペース保護の一形態として付けられる。

ストリング操作を伴う関数は、ストリングのフィルタリングの点でかなり高機能である。たとえば、Prs_AddContact (「Danny Rabbani」) はPrs_AddContact (「Danny Rabbani」) と等価である。この場合、大文字使用は重要である。Prs_DeleteContact (「Danny Rabbani」) はPrs_DeleteContact (「danny raBBanI」) と等価である。Prs_AddContact (「Danny Rabbani」) はPrs-AddContact (「danny rabbani」) と等価ではない。これは、パーサ300が内部的に担当者それぞれ「Danny Rabbani」および「danny rabbani」として表現するためである。しかし、2番目の呼出しは失敗する。これは、パーサ300は、大文字使用だけが異なる2つの担当者 (

またはリスト、プロジェクト、またはキーワード) が同時に存在することを許さないためである。この種のスマート・フィルタリング (先行スペースと後続スペースの除去と、適切な場合には大文字小文字の無視) およびその他の形態の誤りおよび正常性検査が、このDLLの関数のほとんどに適切に適用される (aut

o f i l l 関数のみは大文字小文字の相違を許容する。ホワイト・スペースは相違を生じる)。しかし、クライアントは、少なくともその一部を事前に検査せずにこのような機能に依存し過ぎないことを推奨する。

***** /

```
#ifndef PARSEAPI_H
#define PARSEAPI_H
extern "C" {
```

/*

V i s u a l B a s i c (VB) と通信する場合、128のバッファ・サイズで最大63文字までの長さのストリングが可能であることに留意されたい。これは、VBが常にユニコード文字を使用し、それによってASCII文字を記憶するのに必要なバイト数が事実上2倍になる。ユニコードの変換はVBによって自動的に処理される。

* /

```
#define PRS_BUFFER_SIZE 128
```

```
//Units
```

```
#define PRS_DAYS 1
```

```
#define PRS_WEEKS 2
```

```
#define PRS_MONTHS 3
```

```
#define PRS_YEARS 4
```

```
//Weekdays
```

```
#define PRS_SUN 1
#define PRS_MON 2
#define PRS_TUE 4
#define PRS_WED 8
#define PRS_THU 16
#define PRS_FRI 32
#define PRS_SAT 64
#define PRS_WEEKDAYS (PRS_MON + PRS_TUE + PRS_WED + PRS_THU + PRS_FRI)
#define PRS_WEEKENDS (PRS_SAT + PRS_SUN)
```

/*-----

STRUCT: ParseOutput

目的: ParseOutput structは、パーサ300DLLのクライアントが作成し、使用するためのものである。ParseOutput structは、入力表現(第1のパラメータ)と共にPrs_Parse()関数に第2のパラメータとして渡される。クライアントがPrs_Parse()を呼び出すと、パーサ300は入力表現を解析し、解析の結果をParseOutput structにパッケージ化する。Prs_Parse()を呼び出す前にParseOutputフィールドのいずれも初期設定する必要がない。Prs_Parse()関数は、式のうちの成功裏に解析されたフィールドのみに書き入れ、他のすべてのフィールドをゼロ長の空終了ストリングまたは他の適切な空値に初期設定する。日付解析のみ行いたい場合、クライアントはPrs_Parse()の代わりにPrs_DateParse()を呼び出す必要がある。Prs_DateParse()はPrs_Parse()と同様に機能するが、ParseOutput structの日付と時刻関係のフィールドのみに書き入れられる(キーワード解析および被委任者解析は迂回される)。この両方の関数呼出しはきわめて効率的である。(妥当な範囲で)長くて複雑な入力表現でも一瞬のうちに解析される。

フィールド：sList：空終了ストリングとしてのリスト名。

sProject：空終了ストリングとしてのプロジェクト名。

sContact：空終了ストリングとしての担当者名。一例は、ファースト・ネームの後に単一の空間が続き、その後に姓が続くものである。

sDelegate：委任先の担当者名。

sDate：空終了ストリングとしての日付。月／日／年形式。例：“12／28／1969”。このフィールドは反復事象の開始日付としても機能する。

sTimeBegin：空終了ストリングとしての開始時刻。[時]時：分[分]24時間形式。すなわち、時間数（0～23）、コロン、分数（0～59）の順。例：“3：0”＝午前3時。“23：45”＝午後11時45分。

sTimeEnd：空終了ストリングとしての終了時刻。形式はsTimeBeginと同じ。

sDateEnd：反復事象の終了日。

bRecurring：パーサ300が（単純日付ではなく）反復事象であると判断したことを示すブール・フラグとして機能するバイト・サイズ値。値0（偽）は、反復日付が存在しないことを示す。他のすべての値は真を表す（すなわち反復日付が解析された）。

nFrequency：反復日付の頻度を表す長整数（32ビット）。たとえば、「5年ごと」の頻度は5である。このフィールドの空値は0である。

nUnits：単位（日、週、月、および年）を示すバイト・サイズ値。それぞれ、定数PRS_DAYS、PRS_WEEKS、PRS_MONTHS、およびPRS_YEARSによって定義される。このフィールドの空値は0である。

nWeekdays：曜日（すなわち火曜、月曜、金曜など）を示すバイト・サイズ値。曜日定数（PRS_SUN～PRS_SAT）は、このバイト・フィールドによって最大7つの曜日の任意の組合せをコード化することができるように定義される。特定の曜日がnWeekdays値に含まれているかどうかを調べるには、その曜日の値とnWeekdays値との「AND」を行う（ビット

ト単位のAND演算)だけでよい(たとえばPRS_MON&nWeekdays)。

Null: このフィールドの値は0である。

nWeekdaysPos: 月のうちの当該曜日の序数(第1、第2、第3、第4、第5)位置を示すバイト・サイズ値。たとえば、「毎月の第3火曜日」のnWeekdaysPos値は3になる。このフィールドの空値は0である。

bFYI: sDelegateフィールドが被委任者またはアクション要求キーノートではなくFYIキーノートを表すこと示す。

bNote: ノート・タイプのキーノートであることを示す。

-----*/

```
typedef struct_ParseOutput {
    char sList[PRS_BUFFER_SIZE];
    char sProject[PRS_BUFFER_SIZE];

    char sContact[PRS_BUFFER_SIZE];
    char sDelegate[PRS_BUFFER_SIZE];
    char sDate[PRS_BUFFER_SIZE];
    char sTimeBegin[PRS_BUFFER_SIZE];
    char sTimeEnd[PRS_BUFFER_SIZE];
    char sDateEnd[PRS_BUFFER_SIZE];
    BYTE bRecurring;
    BYTE nUnits;
    BYTE nWeekdays;
    BYTE nWeekdayPos;
    long nFrequency;
    BYTE bFYI;
    BYTE bNote;
} ParseOutput, *ParseOutputPtr;
```

/*-----

STRUCT: KeywordSuggestion

目的: KeywordSuggestion structは、解析DLLのクライアントが作成し、使用するためのものである。KeywordSuggestionデータ構造は、Prs__SuggestKeywords () 関数ファミリの第2のパラメータとして、入力表現 (第1のパラメータ) と共に渡される。クライアントがたとえばPrs__SuggestListKeywords () を呼び出すと、パーサ300は入力表現 (この場合はリストの名前) を分析し、最大3個のキーワード提案をKeywordSuggestionデータ構造にパッケージ化する。キーワード提案関数を呼び出す前にKeywordSuggestionフィールドのいずれも初期設定する必要はない。これらの関数は可能な限り多くのフィールドに (sKeyword1から始まる連続順序で) 書き入れ、他のすべてのフィールドをゼロ長の空終了ストリングに初期設定する。

フィールド: sKeyword1: 空終了ストリングとしての最初のキーワード提案。

sKeyword2: 2番目のキーワード提案。

sKeyword3: 3番目のキーワード提案。

-----*/

```
typedef struct KeywordSuggestion{
    char sKeyword1[PRS_BUFFER_SIZE];
    char sKeyword2[PRS_BUFFER_SIZE];
    char sKeyword3[PRS_BUFFER_SIZE];
}KeywordSuggestion, *KeywordSuggestionPtr;
```

/*-----

関数: Prs__ResetParsedll

目的: Prs__ResetParsedll () 関数は、parsedllをリセットする。Add/Delete関数を使用してDLLに示された情報はすべ

て失われる。日付解析はそのまま完全に機能し続ける。

-----*/

```
void WINAPI EXPORT Prs_ResetParseDll(void);
```

/*-----

関数: Prs_Parse

目的: Prs_Parse () 関数は、リスト、プロジェクト、担当者、被委任者、および日付/時刻の組合せの入力表現を解析する。第1のパラメータは、解析する入力表現を保持する空終了ストリングを指すポインタである。第2のパラメータは、解析結果が入れられるクライアント割り振り ParseOutput データ構造を指すポインタである。メモリの割り振り以外には、ParseOutput データ構造の特別な初期設定は不要である。入力表現から成功裏に導き出すことができない ParseOutput データ構造のフィールドは、適切な空値に設定される。

例: ParseOutput parseResults;

```
Prs_Parse ("Call Brian tomorrow at 6
pm", &parseResults);
```

-----*/

```
void WINAPI EXPORT Prs_Parse(const char FAR* sInputExpr, ParseOutput FAR
* pParseOutput);
```

/*-----

関数: Prs_DateParse

目的: Prs_DateParse () 関数は、日付と時刻（または期間）の入力表現を解析する。第1のパラメータは、解析する入力表現を保持する空終了ストリングを指すポインタである。第2のパラメータは、解析結果が入れられ

るクライアント割振り ParseOutput struct を指すポインタである。メモリの割振り以外は、ParseOutput データ構造の特別な初期設定は不要である。入力表現から成功裏に導き出すことができない ParseOutput データ構造のフィールドは、適切な空値に設定される。

例: ParseOutput parseResults;

```
Prs_DateParse ("Call Tony next Friday", &parseResults);
```

-----*/

```
void WINAPI EXPORT Prs_DateParse(const char FAR* sInputExpr, ParseOutput
FAR* pParseOutput);
```

/*-----

関数: Prs_SuggestListKeywords,

```
Prs_SuggestProjectKeywords
```

```
Prs_SuggestContactKeywords
```

目的: 与えられたリスト、プロジェクト、または担当者のために最大3個のキーワードを提案する。パーサは、すでに使用されているキーワードを提案しない。

例: KeywordSuggestion suggestion;

```
Prs_SuggestContactKeywords ("Danny Rabbani", &suggestion);
```

-----*/


```
void WINAPI EXPORT Prs_SuggestListKeywords(const char FAR* sList,
      KeywordSuggestion FAR* pSuggestion);
```

```
void WINAPI EXPORT Prs_SuggestProjectKeywords(const char FAR* sProject,
      KeywordSuggestion FAR* pSuggestion);
```

```
void WINAPI EXPORT Prs_SuggestContactKeywords(const char FAR* sContact,
      KeywordSuggestion FAR* pSuggestion);
```

```
/*-----
```

関数: Prs__AddKeyword/Prs__DeleteKeyword/

Prs__RenameKeyword

目的: キーワードの追加/削除/名前変更を行う。これらの関数は、初期設定中またはデータベースを更新する際に呼び出す必要がある。

例: Prs__AddKeyword(“na”, “Next Actions”, “”, “”);

Prs__DeleteKeyword(“na”);

Prs__RenameKeyword(“na”, “actions”);

注: Prs__AddKeywordは、そのキーワードがすでに存在する場合や、非空リンクのいずれかが存在しない場合、またはすべてのリンクが空の場合、偽を返す。クライアントは必ず、対応するリスト、プロジェクト、または担当者がすでにパーサ300に加えられた後で初めて、この関数を呼び出すようにしなければならない。キーワードを名前変更するには、クライアントはPrs__RenameKeyword()を呼び出すだけでよい。いかなる種類の大文字使用も無視される。

```

----- */

BOOL WINAPI EXPORT Prs_AddKeyword(const char FAR* sKeyword,

    const char FAR* sList,
    const char FAR* sProject,
    const char FAR* sContact);

BOOL WINAPI EXPORT Prs_DeleteKeyword(const char FAR* sKeyword);
BOOL WINAPI EXPORT Prs_RenameKeyword(const char FAR* sOldKeyword, const
char FAR* sNewKeyword);

```

```

/*-----
関数：Prs__AddList／Prs__DeleteList／Prs__Re
nameList

```

目的：リストの追加／削除／名前変更を行う。これらの関数は、オブジェクト・データベース850を更新するときには必ず呼び出す必要がある。

```

例：Prs__AddList ( "Next Calls" ) ;
    Prs__DeleteList ( "Next Calls" ) ;
    Prs__RenameList ( "Next Calls", "My Ne
xt Calls" ) ;

```

注：Prs__AddListは、（大文字使用が異なる場合であっても）リストがすでに存在する場合、偽を返す。Prs__DeleteListは、リストが存在しない場合、偽を返す。Prs__DeleteListまたはPrs__RenameListが成功した場合、そのリストにリンクされているキーワードは

すべて、必要に応じて自動的に除去または更新される。

-----*/

```

BOOL WINAPI EXPORT Prs_AddList(const char FAR* sList);
BOOL WINAPI EXPORT Prs_DeleteList(const char FAR* sList);
BOOL WINAPI EXPORT Prs_RenameList(const char FAR* sOldList, const char FAR* sNewList);

```

/*-----

関数: Prs_AddProject / Prs_DeleteProject /
Prs_RenameProject

目的: プロジェクトの追加/削除/名前変更を行う。これらの関数は、オブジェクト・データベース850を更新するときには必ず呼び出す必要がある。

```

例: Prs_AddProject ( "Learn Spanish" );
    Prs_DeleteProject ( "Learn Spanish" );
    Prs_RenameProject ( "Learn Spanish",
        "Learn French" );

```

注: Prs_AddList の注を参照。

-----*/

```

BOOL WINAPI EXPORT Prs_AddProject(const char FAR* sProject);
BOOL WINAPI EXPORT Prs_DeleteProject(const char FAR* sProject);
BOOL WINAPI EXPORT Prs_RenameProject(const char FAR* sOldProject, const
char FAR* sNewProject);

```

/*-----

関数: Prs_AddContact / Prs_DeleteContact /
Prs_RenameContact

目的: 担当者の追加/削除/名前変更を行う。これらの関数は、オブジェクト・

データベース850を更新するときには必ず呼び出す必要がある。

```
例: Prs__AddContact ( "Danny Rabbani" );
    Prs__DeleteContact ( "Danny Rabbani" );
    Prs__RenameContact ( "Danny Rabbani",
    "Dan Rabbani" );
```

注: Prs__AddList の注を参照。

-----*/

```
BOOL WINAPI EXPORT Prs_AddContact(const char FAR* sContal);
BOOL WINAPI EXPORT Prs_DeleteContact(const char FAR* sContact);
BOOL WINAPI EXPORT Prs_RenameContact(const char FAR* sOldContact, const
char FAR* sNewContact);
```

/*-----*/

/* AutoFill 関数 */

/*-----*/

/*-----

関数: Prs__AutoFillList, Prs__AutoFillProject, Prs__AutoFillContact

目的: サブstringの完成を返す。これらの関数は、第1のパラメータとしてプレフィックス・stringをとり、バッファ (第2のパラメータ) をプレフィックスの完成で満たし、真を返す。完成が存在しない場合、これらの関数は偽を返す (下記の「注」を参照)。これらの関数は、プレフィックス・string (第1のパラメータ) に関しては大文字小文字を区別しない。これには、「d」のようなプレフィックスだけでなく、「D」プレフィックスも、「Danny Rabbani」に展開されるという利点がある。

注: 完成が存在しない場合、これらの関数は偽を返し、バッファ (第2のパラメータ) は修正されない。

```
例: Prs__AddProject ( "Grasshopper" ) char
```

```

sProject[PRS_BUFFER_SIZE];

Prs_AutoFillProject("gra", sProject)

;

AffxMessageBox(sProject); // "Grasshopper"
per"を出力する。
-----*/

BOOL WINAPI EXPORT Prs_AutoFillList(const char FAR* sListPrefix, char FAR* sListBuffer);
BOOL WINAPI EXPORT Prs_AutoFillProject(const char FAR* sProjectPrefix, char FAR* sProjectBuffer);
BOOL WINAPI EXPORT Prs_AutoFillContact(const char FAR* sContactPrefix, char FAR* sContactBuffer);

}

#endif /*PARSEAPI_H*/

```

サンプル電子メール・メッセージ

本発明を使用して、前述のような方式で入力キーノートの受信と解析を行うことができる。さらに、本発明は、各ユーザが本発明の機能を備えているか否かを問わず、他のユーザとのコラボレーションをサポートする。キーノートの受信者は本発明の機能を備えていない場合があるため、本発明は従来の電子メール・メッセージを自動的にフォーマットし、それをキーノートの受信者または被委任者に送信することができる。前述のようにパーサ300を使用して、本発明は入力キーノートを受け取り、その入力キーノート内の一致したキーワードに関連づけられた情報から、従来の電子メール・メッセージを構築する。さらに、本発明は、電子メール・メッセージのスクリプト化（すなわちコード化）版も構築する。電子メール・メッセージのスクリプト化版は、キーノートの受信者または被委任者が本発明の機能を有する場合に、メッセージを自動的に処理することができる。

ように構造化された形式にフォーマットされる。受信者または被委任者が本発明の機能を有する場合、受信側でキーノートのスクリプト化版を解釈し、キーノートとして自動的に処理することができる。したがって、本発明は、従来のテキスト形式とスクリプト形式の両方を同じ自動生成電子メッセージとして結合して電子メッセージを構築することにより、そのような電子メール・メッセージの受信側の誰とでもキーノート・コラボレーションを行えるようにする。

以下のサンプル自動生成キーノート電子メール・メッセージでは、以下の入力キーノートが与えられた場合に本発明により自動的に作成される二重形式電子メッセージ構造を例示する。

```
Jim, please add this sample KeyNote  
  
to the KeyNote/parser patent tomorrow.  
Call Tom if you have questions. Thanks!
```

上記の入力キーノートから本発明により生成されるサンプル電子メール・メッセージは以下の通りである。以下のメッセージの最初の部分は、受信者が本発明の機能がなくても読める従来の電子メール形式（ASCII）を表す。「dreamldbssbt dwr v k val」というテキスト・ストリングのメッセージが2回目に出現する箇所から始まるメッセージの2番目の部分は、受信者が本発明の機能を使用して処理することができるメッセージのスクリプト化版を表す。したがって、このサンプル電子メール・メッセージは、本発明の二重形式電子メール・メッセージ生成機能を例示するものである。本発明のコラボレーション機能の詳細な説明は、本特許出願の後の部分に記載する。

--begin sample electronic mail message--

This is a multipart message in MIME format.

--dreldbssbtdwrvkval

<other content transfer encoding type being used,
probably quoted-printable or 7-bit>

Jim, please add this sample KeyNote to the
KeyNote/parser patent tomorrow. Call Tom if you have
questions. Thanks!

This is an ActionRequest for Jim Salter from Dennis
Buchheim.

It relates to:

Project: Patents
Contact: Tom Hagan
Due Date: 2/6/97

PROJECT INFORMATION

Name: Patents
Due: 2/5/97
Outcome: Successfully defend Actioneer's inventions.

CONTACT INFORMATION

Tom Hagan
Chairman
Actioneer, Inc.
539 Bryant St.
San Francisco, CA 94107
USA
(415) 555-1212 (Work)
hagan@actioneer.com (Internet)

--dreldbssbtdwrvkval

<other content transfer encoding type being used,
probably quoted-printable or 7-bit>

X-Keynote-Delegate: TRUE
BEGIN:VCARD
X-Version:1.0.0.0.0
X-Type:ENVELOPE

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:160

X-GUID:00000000-0000-0000-0000-000000000000

X-ID:0

X-Type:PERSON

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:1

X-Salutation:

N:Buchheim;Dennis

X-MiddleName:

X-Type:EMAIL_ADDR

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:2

X-EmailAddrType:+//ISBN 1-887687-00-

9::versit::PDI//INTERNET

EMAIL;INTERNET:buchheim

X-List:1

X-Type:PERSON

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:1

X-Salutation:

X-FamilyName:Buchheim

X-GivenName:Dennis

X-MiddleName:

X-Type:EMAIL_ADDR

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:2

X-EmailAddrType:+//ISBN 1-887687-00-

9::versit::PDI//INTERNET

EMAIL;INTERNET:buchheim
X-List:1
X-Type:EMAIL_ADDR
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:136
X-EmailAddrType:+//ISBN 1-887687-00-
9::versit::PDI//INTERNET
EMAIL;INTERNET:jim_salter@bstz.com
X-List:3
X-Type:EMAIL_ADDR
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:126
X-EmailAddrType:+//ISBN 1-887687-00-
9::versit::PDI//INTERNET
EMAIL;INTERNET:buchheim@actioneer.com
X-Type:EMAIL_ADDR
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:151
X-EmailAddrType:+//ISBN 1-887687-00-
9::versit::PDI//INTERNET
EMAIL;INTERNET:hagan@actioneer.com
X-Type:EMAIL_ADDR
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:143
X-EmailAddrType:+//ISBN 1-887687-00-
9::versit::PDI//INTERNET
EMAIL;INTERNET:judith_szepesi@bstz.com
X-Type:LIST_ITEM
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:157

X-ItemType:ACTION

X-Complete:OPEN

X-Collaboration:REQUESTOUT

X-Priority:0

X-List:1

X-BodyText:Jim, please add this sample KeyNote to the
KeyNote/parser patent tomorrow. Call Tom if you have
questions. Thanks!

X-List:1

X-ReplyText:

X-Date:35466.812778

X-List:1

X-Type:PERSON

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:130

X-Salutation:

X-FamilyName:Salter

X-GiveName:Jim

X-MiddleName:

X-List:1

X-Type:PROJECT

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:153

X-ProjName:Patents

X-Outcome:Successfully defend Actioneer's inventions.

X-Type:SIMPLE_DATE

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:156

X-DateType:REFONLY

X-SimpleDate:-

1. 35466. 000000. 35468. 000000, 0. 000000, 0. 000000

X-List:1

X-Type:PROJECT

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:153

X-ProjName:Patents

X-List:1

X-Type:PERSON

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:144

X-Salutation:

X-FamilyName:Hagan

X-GivenName:Tom

X-MiddleName:

ORG:Actioneer, Inc. ;

TITLE:Chairman

X-List:1

X-Type:ADDRESS

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:152

X-AddrType:+//ISBN 1-887687-00-9::versit::PBI//WORK

ADDR:; 539 Bryant St. ;San francisco;CA;94;07;USA

X-List:1

X-Type:PHONE

X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3

X-ID:150

TEL;WORK:(415) 555-1212

X-List:1
X-Type:EMAIL_ADDR
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:151
X-EmailAddrType:+//ISBN 1-887687-00-
9::versit::PDI//INTERNET
EMAIL;INTERNET:hagan@actioneer.com
X-List:1
X-Type:PERSON
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:144
X-Salutation:
X-FamilyName:Hagan
X-GivenName:Tom
X-MiddleName:
X-Type:SIMPLE_DATE
X-GUID:1de904e1-7f86-11d0-b001-00c026303ba3
X-ID:159
X-DateType:REFONLY
X-SimpleDate:-1, 35467. 000000, 0. 000000, 0. 000000, 0. 000000
END:VCARD
--dreldbssbtdwrvkval--
--end of simple electronic mail message--

以上、コンピュータ・システムのための自然言語ベースの情報編成およびコラボレーションツールについて説明した。本明細書では、本発明について特定の例に関して説明したが、本発明は特定の例の使用には限定されず、他の実施形態にも拡大可能であることは、当業者には明らかであろう。本発明は、以下の請求の範囲で規定されている他のすべての実施形態を含むものと意図される。

【図1】

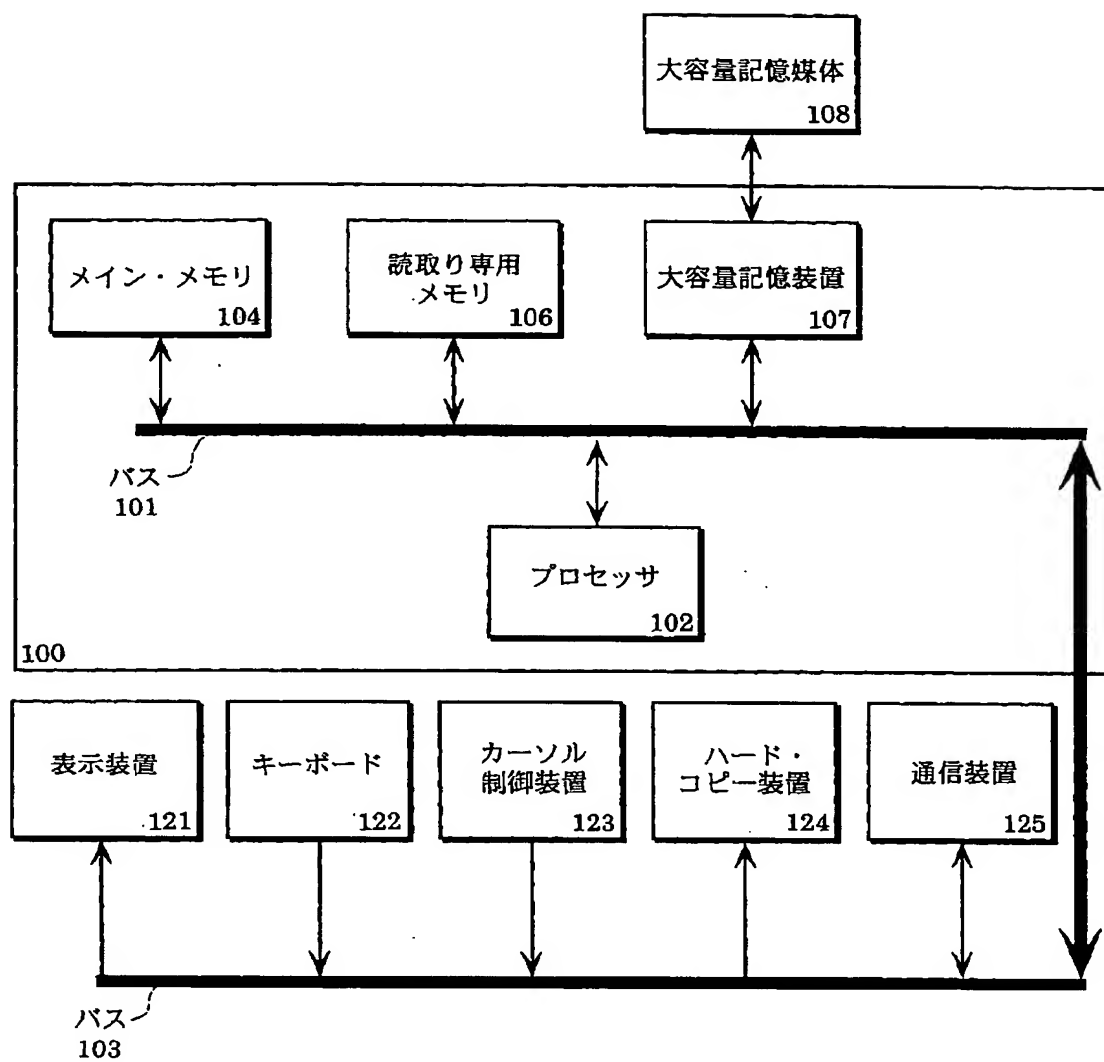


FIG. 1

【図2】

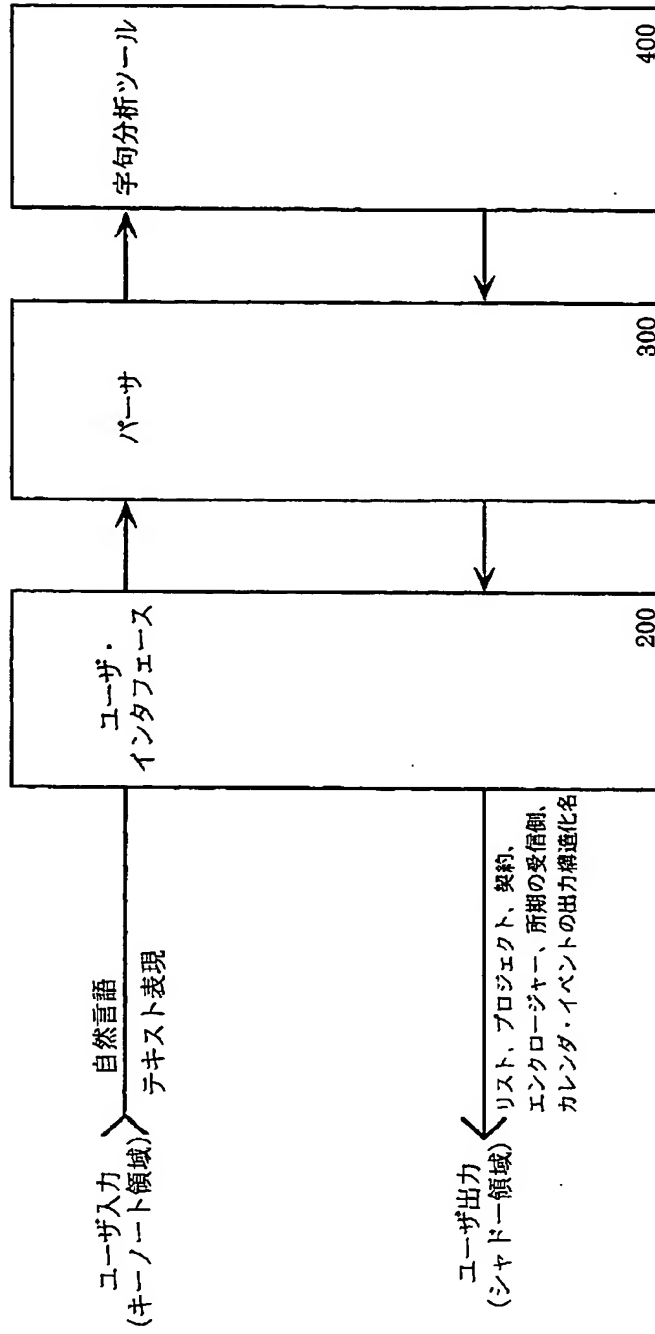


FIG. 2

【図3】

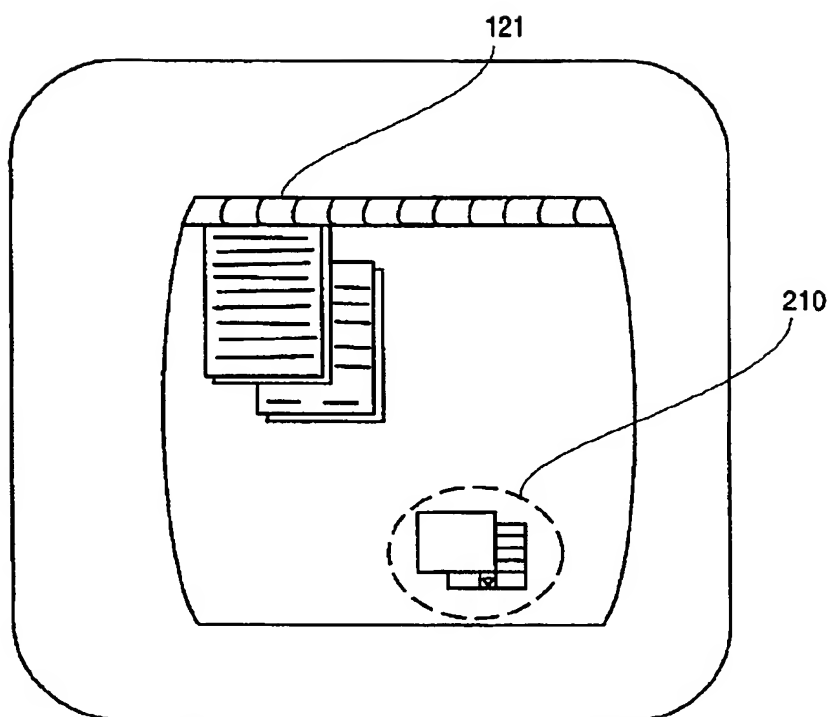
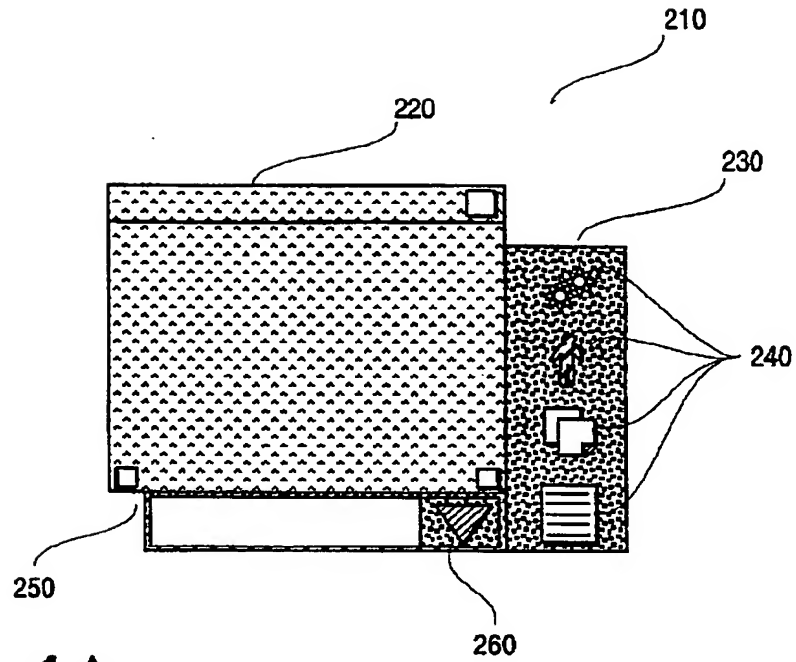
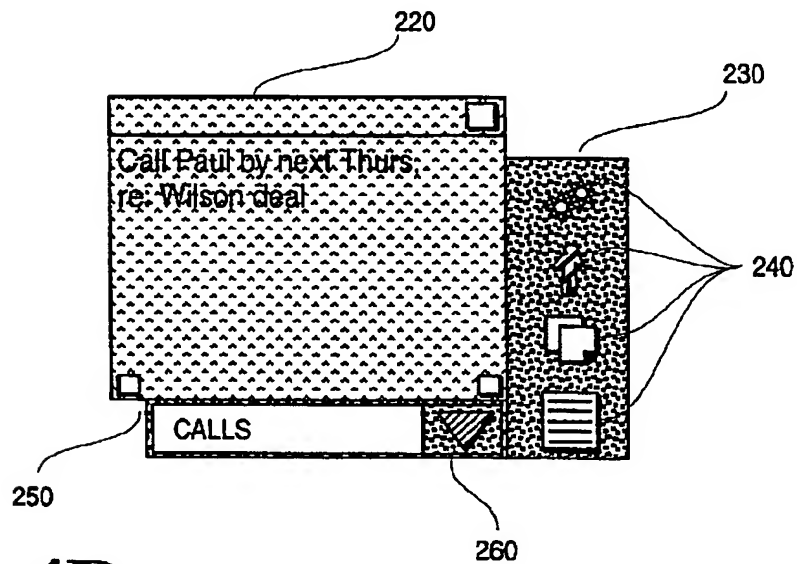
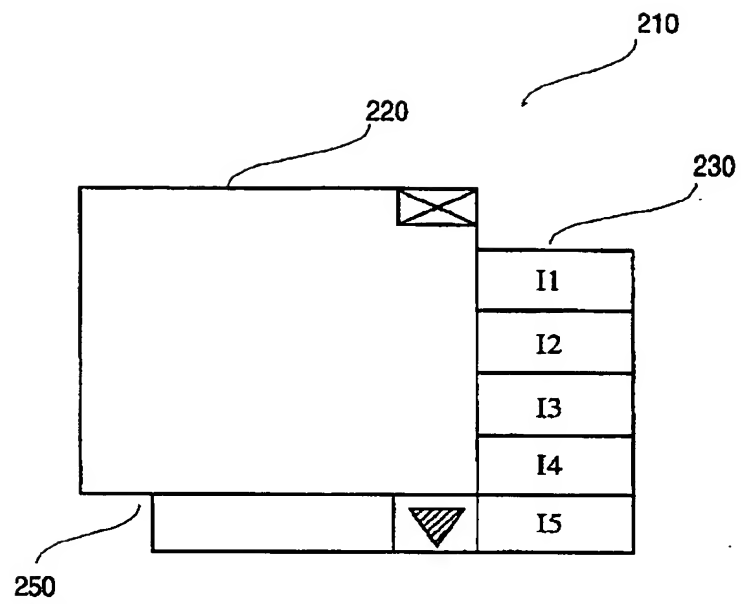


FIG. 3

【図4】

**FIG. 4A****FIG. 4B**

【図5】

**FIG. 5**

【図6】

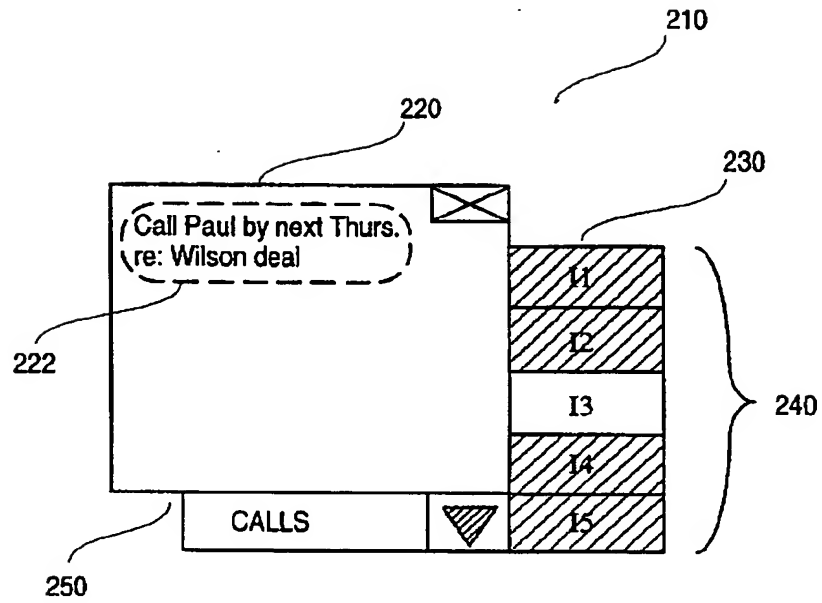


FIG. 6

【図7】

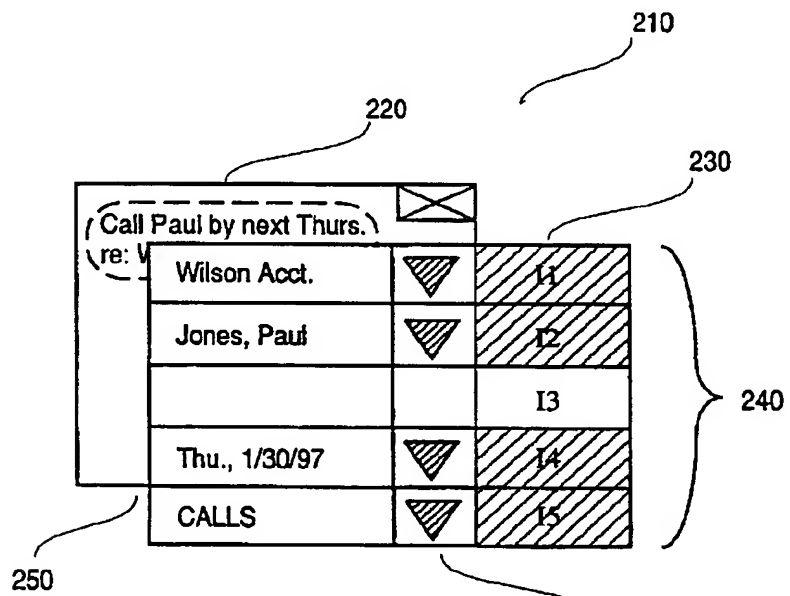


FIG. 7

【図8】

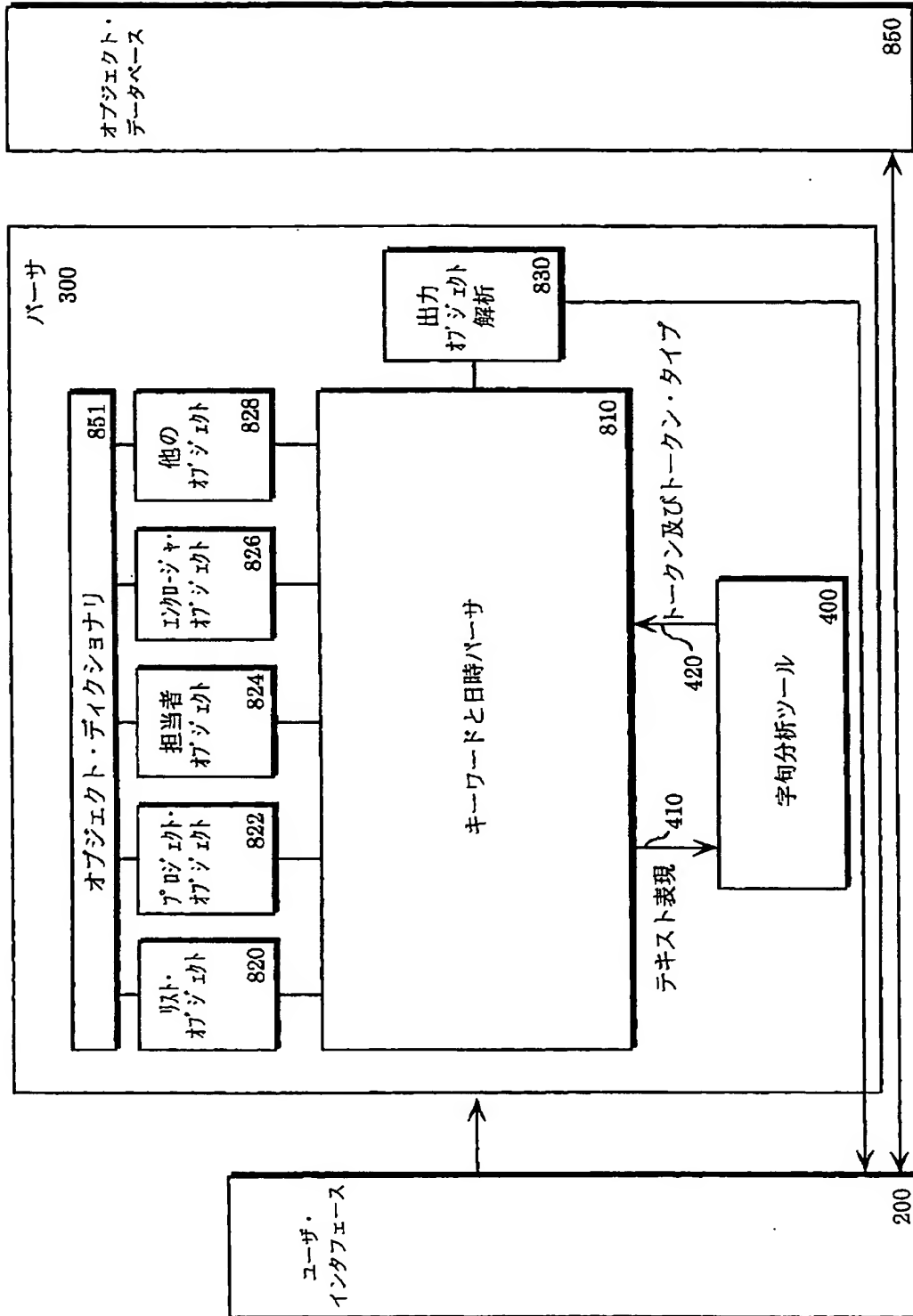


FIG. 8

【図9】

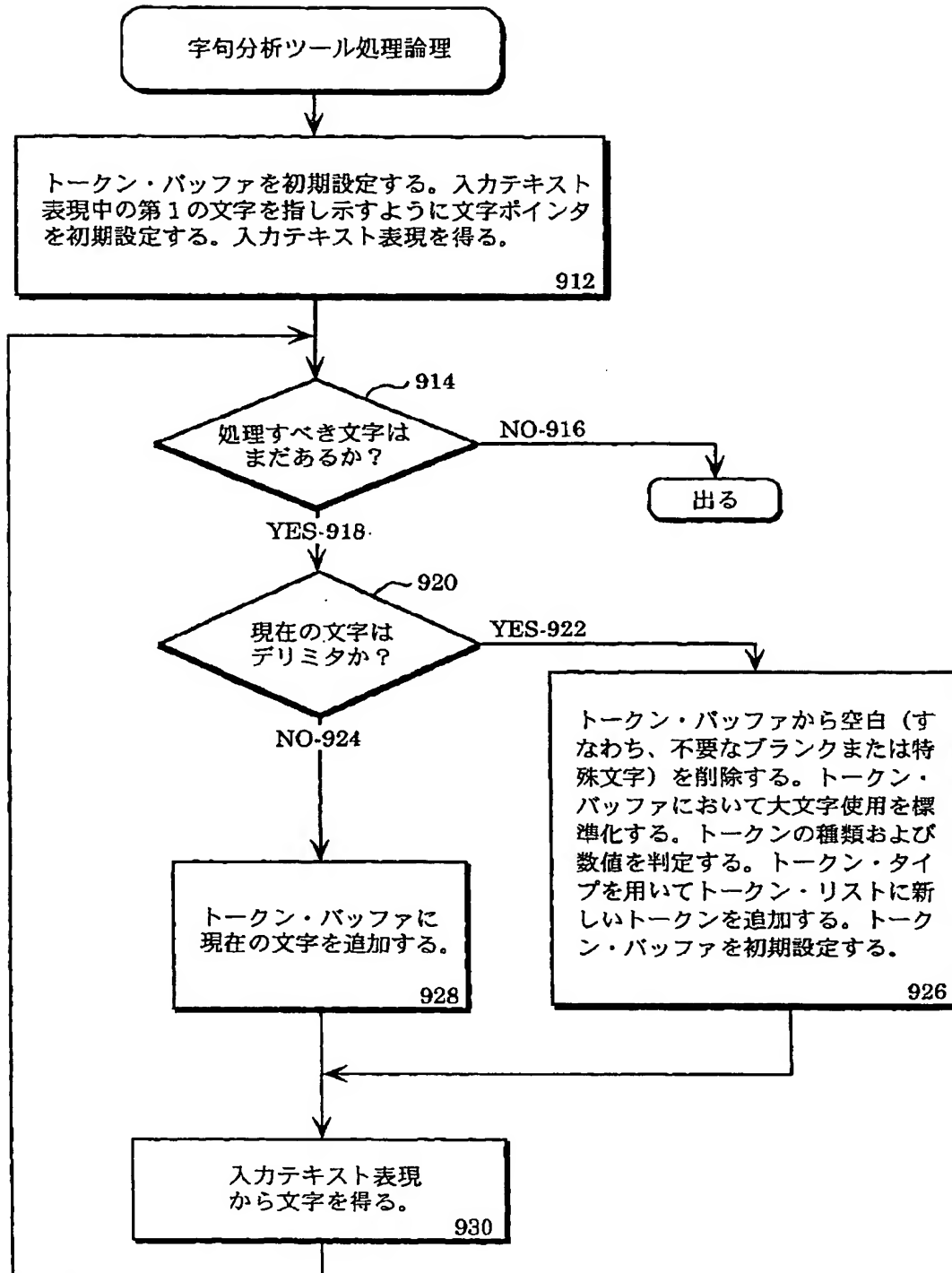


FIG. 9

【図10】

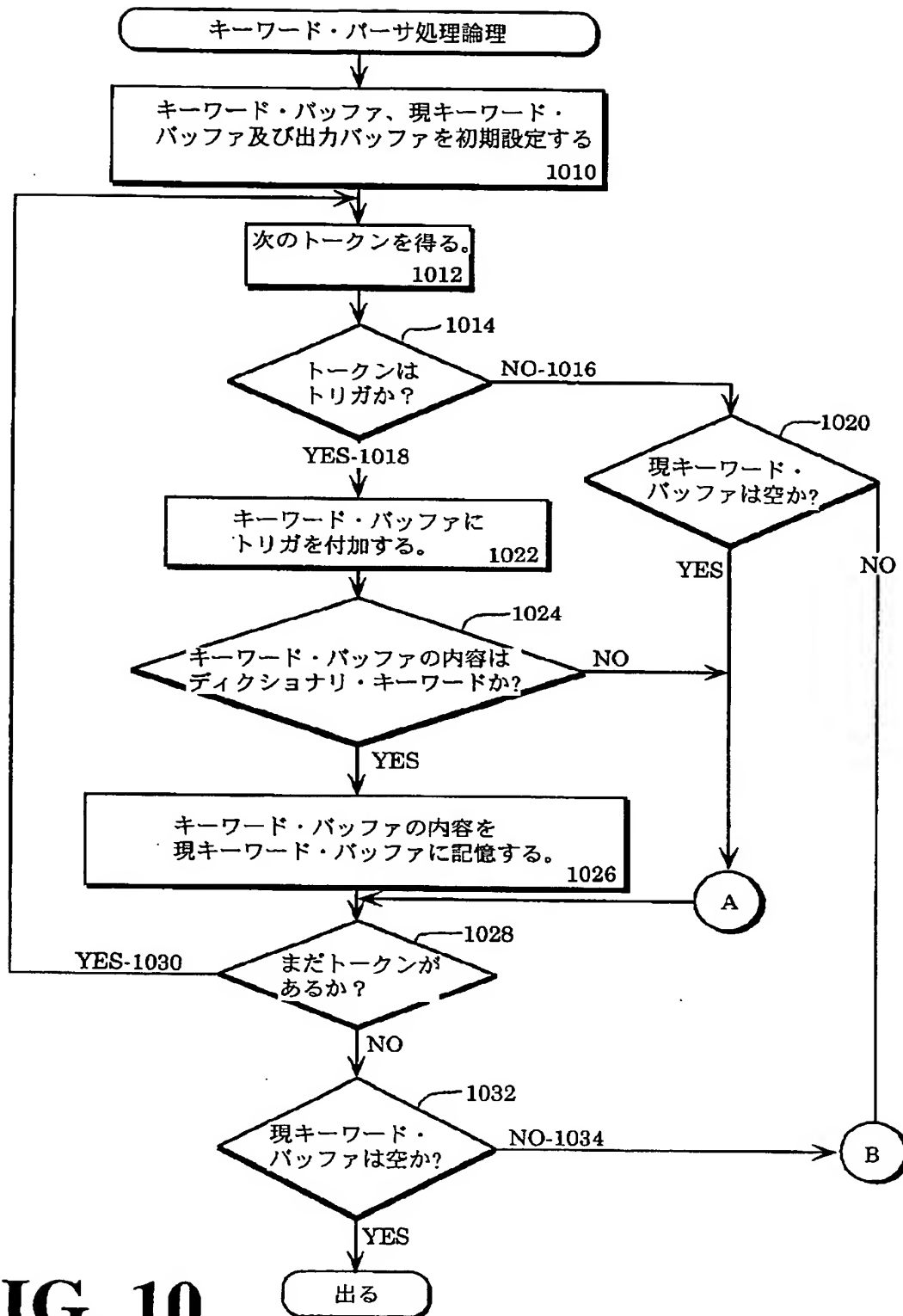


FIG. 10

【図11】

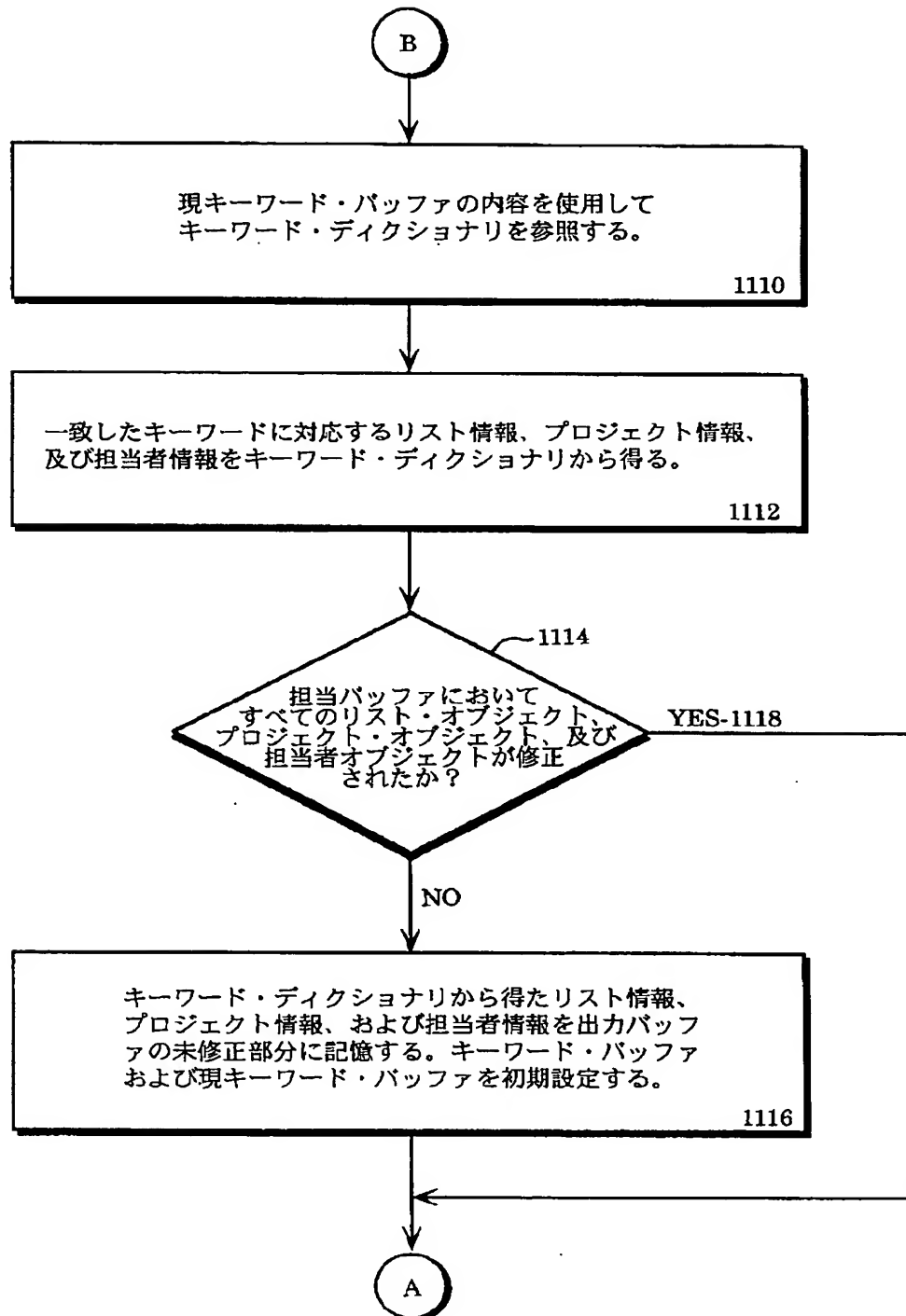


FIG. 11

【図12】

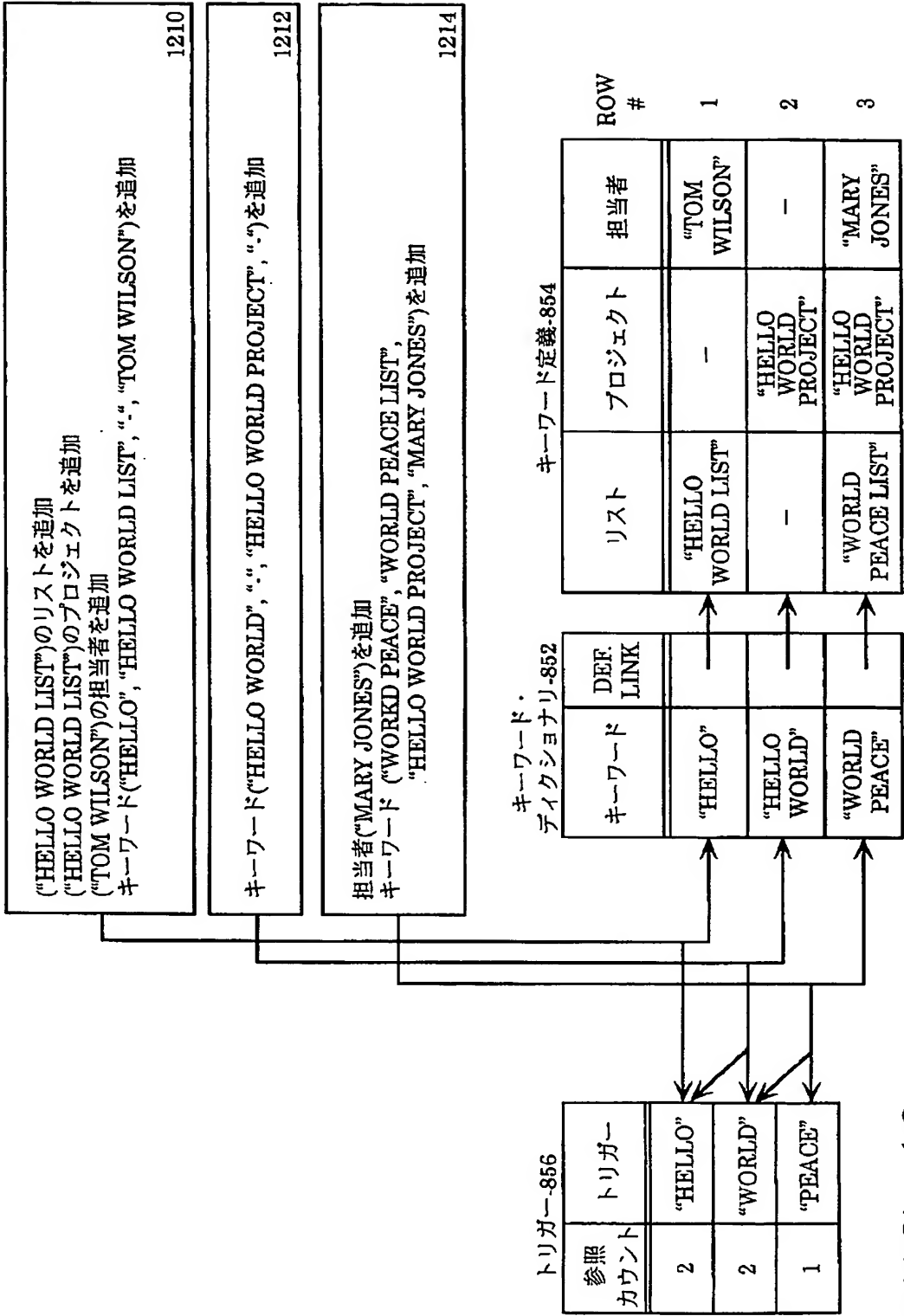


FIG. 12

【図13】

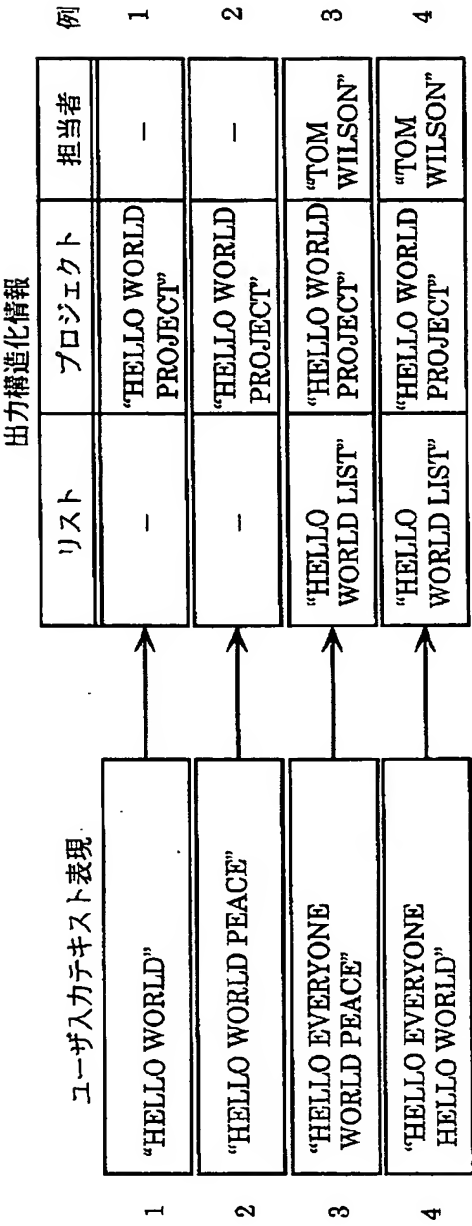


FIG. 13

【図14】

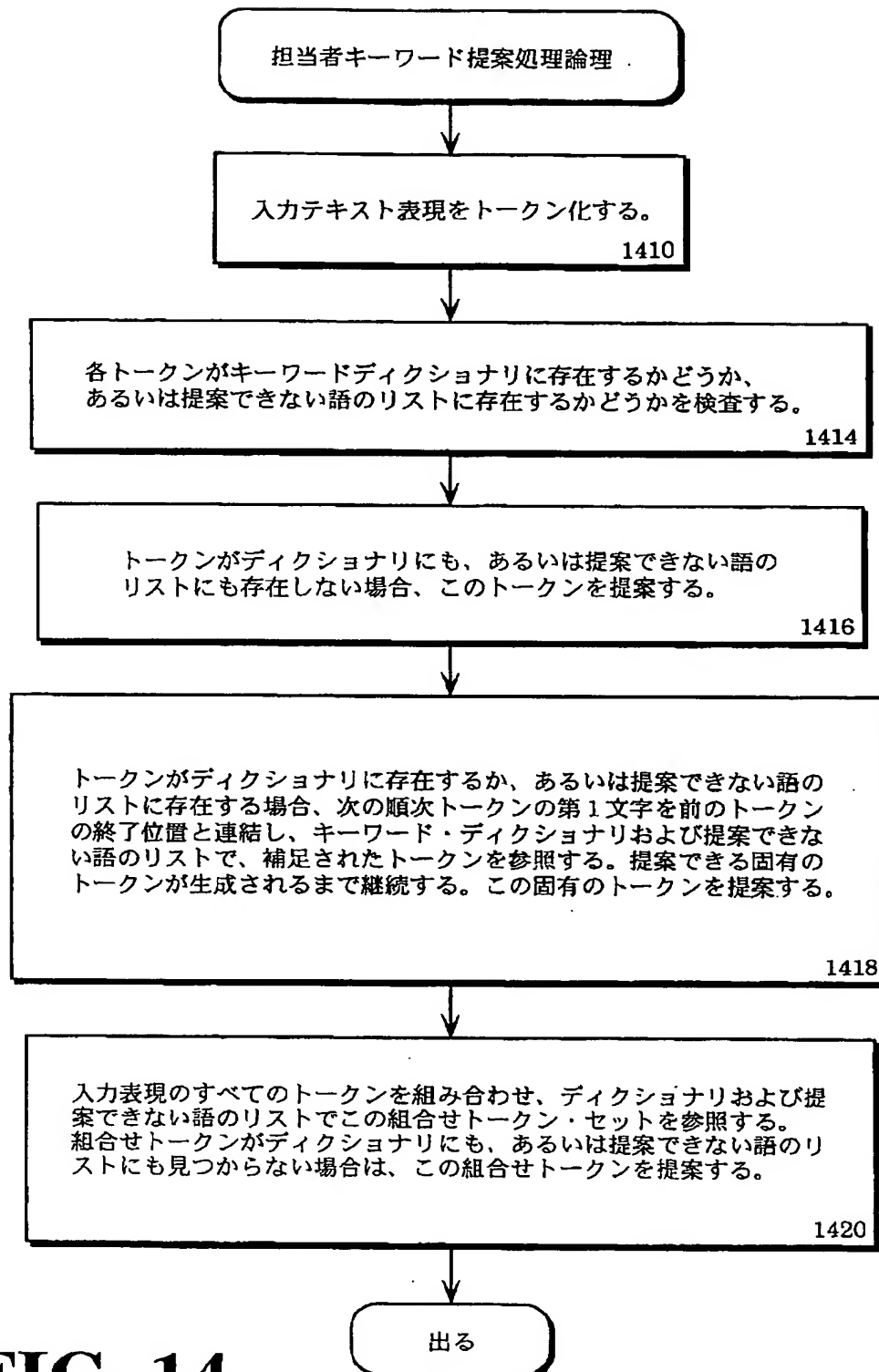


FIG. 14

【図15】

オブジェクト・タイプ・テーブル

キー	タイプ	説明
0	エンベロープ	送るキーノートの入れ物
1	リスト項目	実際のキーノート本体と状況情報
2	プロジェクト	アクション・プロジェクト
3	単純日付	非反復 (期限) 日
4	反復日付	反復 (期限) 日
5	個人	個人連絡先
6	リスト	アクション・アクション・リストまたはメモ・リスト
7	被委任者	協同ノートの被委任者
8	FYI 受信者	参考のためのみのノートの送り先の相手
9	付加プロジェクト	ノートに付加されたプロジェクト
10	付加個人	ノートに付加された個人
11	キーワード	アクション・パーサがノートのテキストを解釈するために使用する語
12	物理アドレス	自宅、職場などのアドレス
13	Eメール・アドレス	インターネット、エクスチェンジなどのEメール・アドレス
14	電話番号	自宅、職場などの電話番号
15	Eメール送信先	ノートの送信先のEメール・アドレス (被委任者)
16	Eメール参考上方	ノートを「cc」として送信する送信先Eメール・アドレス (FYI 受信者)
17	エンベロープ作成者	特定のエンベロープを元々作成した人
18	エンベロープ送信者	特定のエンベロープを現行受信者に送った人

FIG. 15

【図16】

オブジェクト・アソシエーション・テーブル

タイプ	関連づけられたテーブル
エンベロープ	エンベロープ群
リスト項目	リスト項目群
プロジェクト	プロジェクト群
単純日付	単純日付群
反復日付	反復日付群
個人	個人群
リスト	リスト群
被委任者	個人群
F Y I 受信者	個人群
付加プロジェクト	プロジェクト群
付加個人	個人群
キーワード	キーワード群
物理アドレス	物理アドレス群
Eメール・アドレス	Eメール・アドレス群
電話番号	電話番号群
Eメール送信先	Eメール・アドレス群
EメールF Y I	Eメール・アドレス群
エンベロープ作成者	個人群
エンベロープ送信者	個人群

FIG. 16

【図17】

プロジェクト群 (タイプ=2)

キー	名称
201	特許

サンプル・データベース

単純日付群 (3)

キー	日付
202	1997年2月7日木曜

個人群 (5)

キー	名前
203	BRIAN SMIGA
204	JIM SALTER
205	TOM HAGAN
206	DENNIS BUCHEIM

リスト群 (6)

キー	名称
207	電話

キーワード群 (11)

キー	キーワード
208	電話
209	特許

Eメール・アドレス群 (13)

キー	アドレス
210	smiga@actioneer.com
211	hagan@actioneer.com

リンク群

キー1	タイプ1	オブジェクト記述1	キー2	タイプ2	オブジェクト記述2
207	6	「電話」リスト	208	11	「電話」キーワード
201	2	「特許」プロジェクト	209	11	「特許」キーワード
203	5	「BRIAN SMIGA」個人	210	13	「smiga@actioneer.com」 Eメール・アドレス
205	5	「TOM HAGAN」個人	211	13	「hagan@actioneer.com」 Eメール・アドレス

FIG. 17

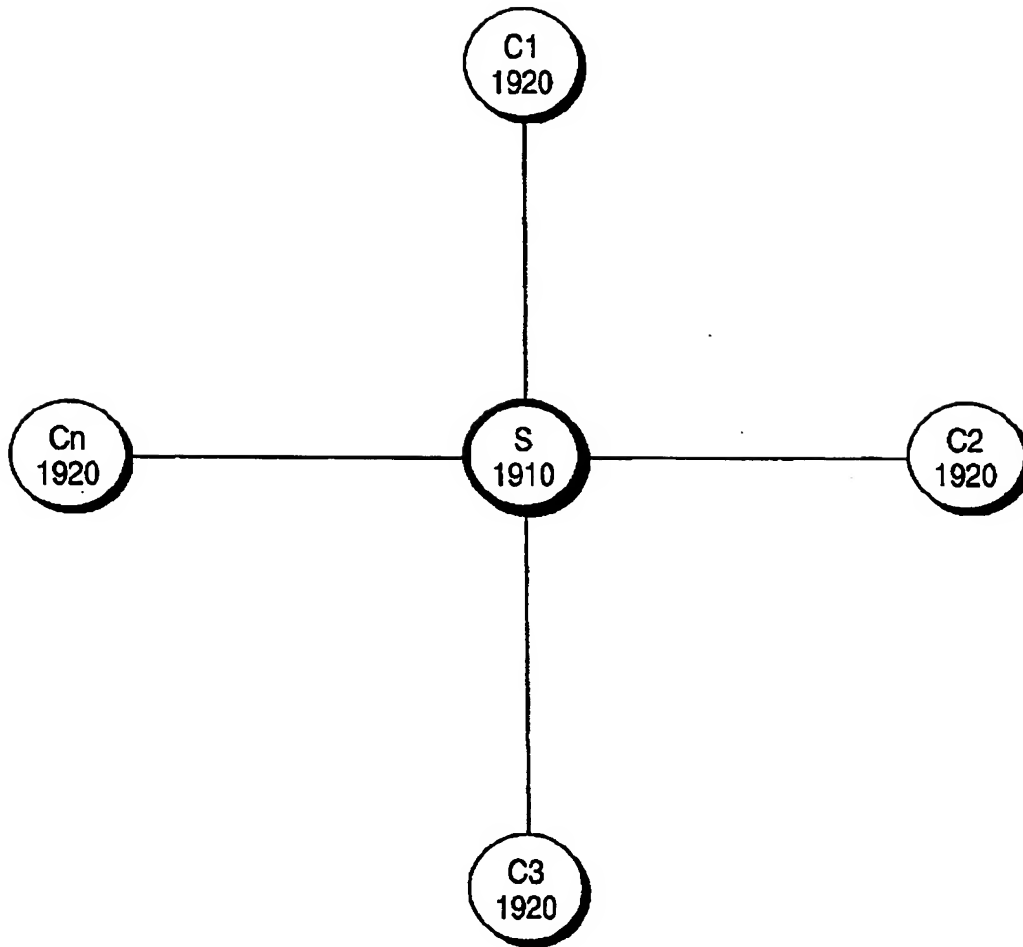
【図18】

サンプル・リンク・テーブル

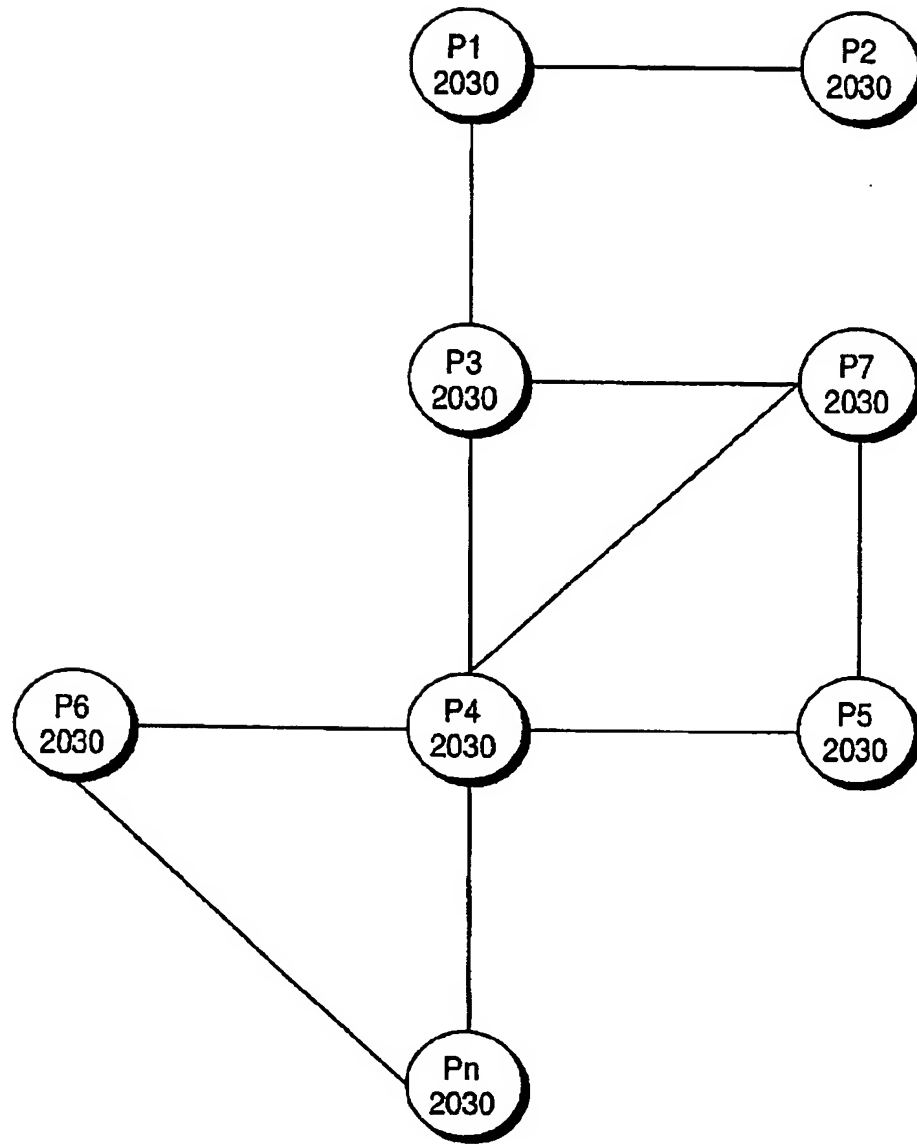
キー1	タイプ1	オブジェクト記述1	キー	タイプ	オブジェクト記述2
212	0	エンベロープ	213	1	リスト項目 (実際のノート)
212	0	エンベロープ	206	17	「デニス・バックハイム」 エンベロープ作成者
212	0	エンベロープ	206	18	「デニス・バックハイム」 エンベロープ送信者
212	0	エンベロープ	210	15	「emiga@actioneer.com」 Eメール宛先
212	0	エンベロープ	211	16	「Hagan@actioneer.com」 EメールFYI
213	1	リスト項目	201	2	「特許」プロジェクト
213	1	リスト項目	202	3	「2/7/97」単純日付
213	1	リスト項目	203	5	「ブライアン・スミガ」個人
213	1	リスト項目	204	5	「ジム・ソルター」個人
213	1	リスト項目	207	6	「電話」リスト
213	1	リスト項目	203	7	「ブライアン・スミガ」被委任者
213	1	リスト項目	205	8	「トム・ハガン」FYI
213	1	リスト項目	201	9	「特許」付加プロジェクト
213	1	リスト項目	204	10	「ジム・ソルター」付加個人

FIG. 18

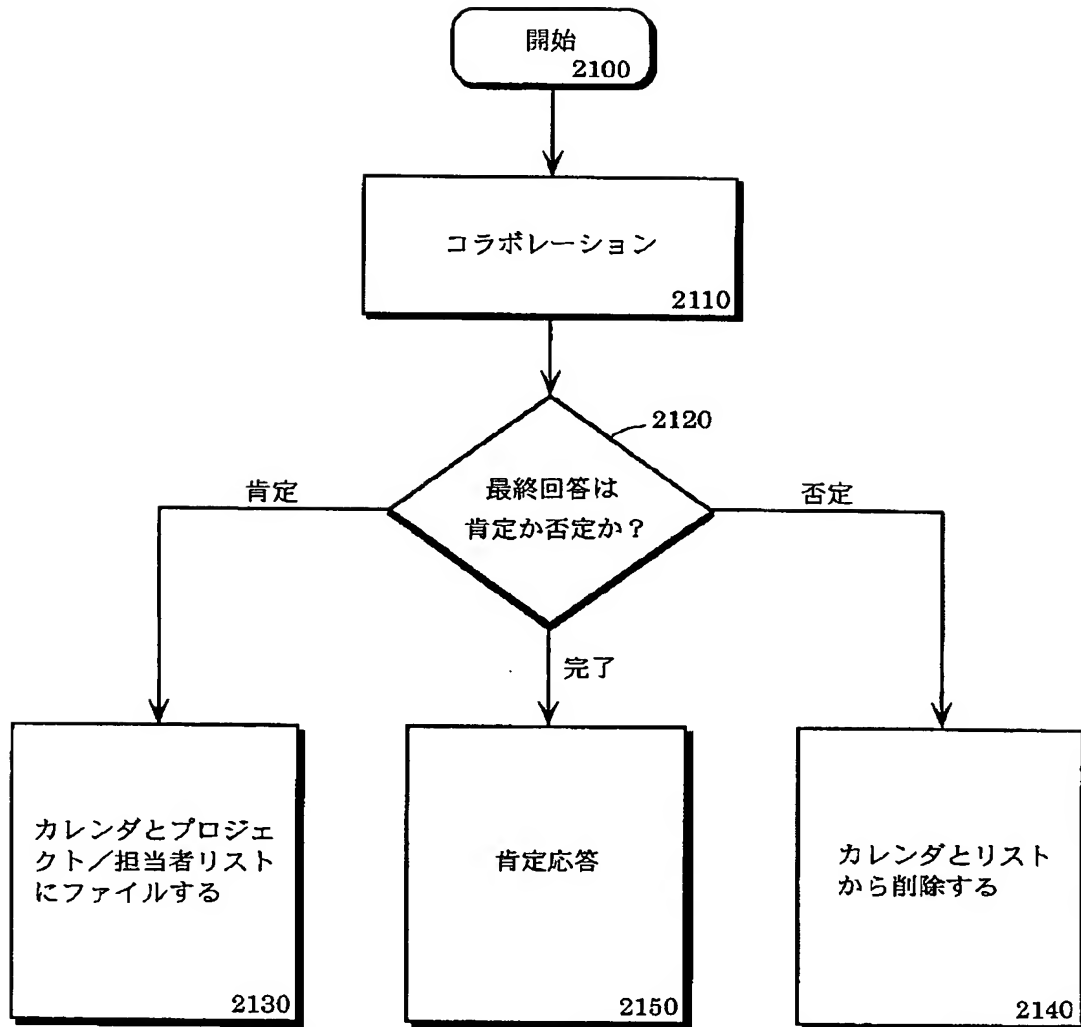
【図19】

**FIG. 19**

【図 20】

**FIG. 20**

【図21】

**FIG. 21**

【図22】

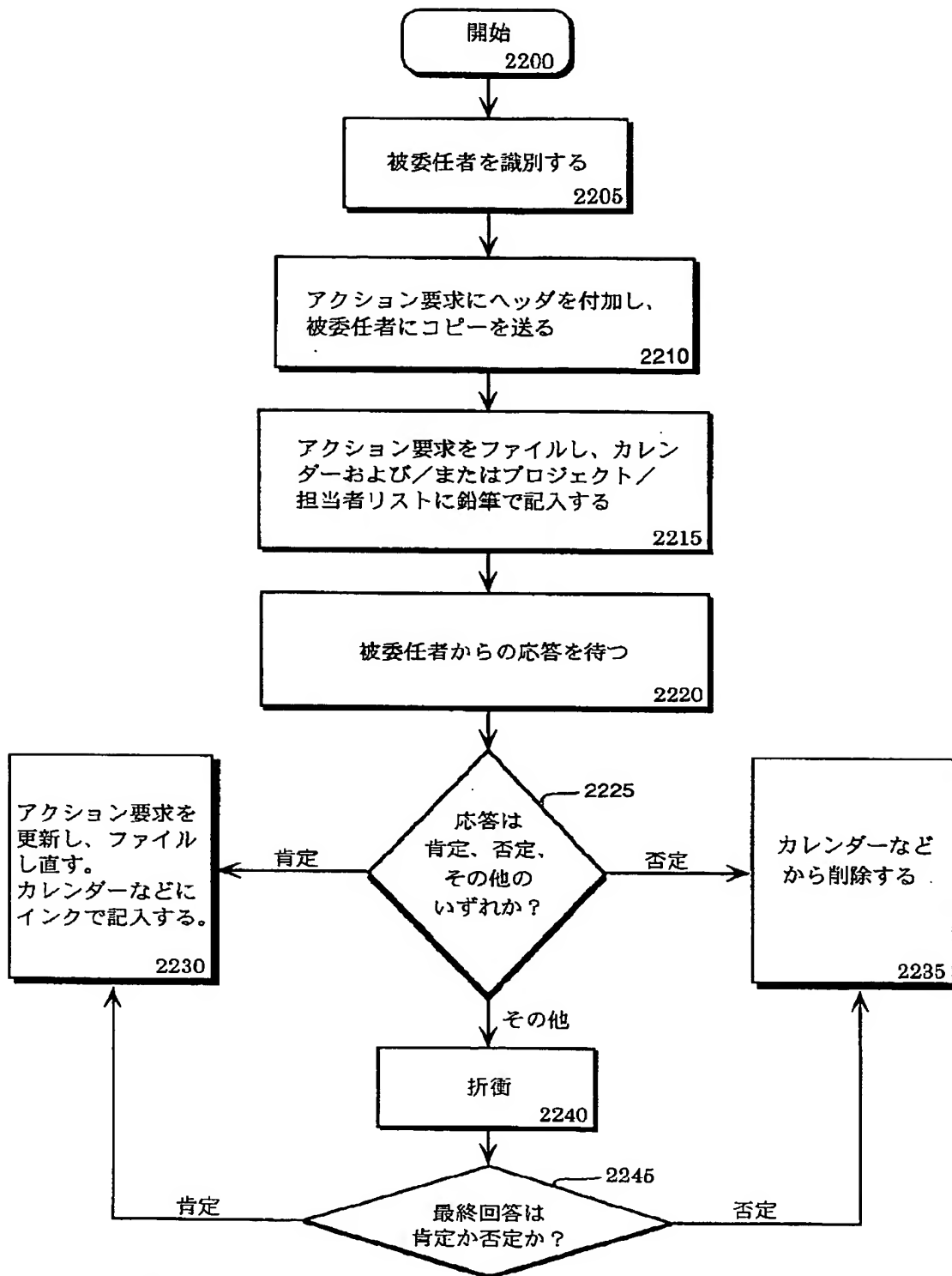


FIG. 22A

【図22】

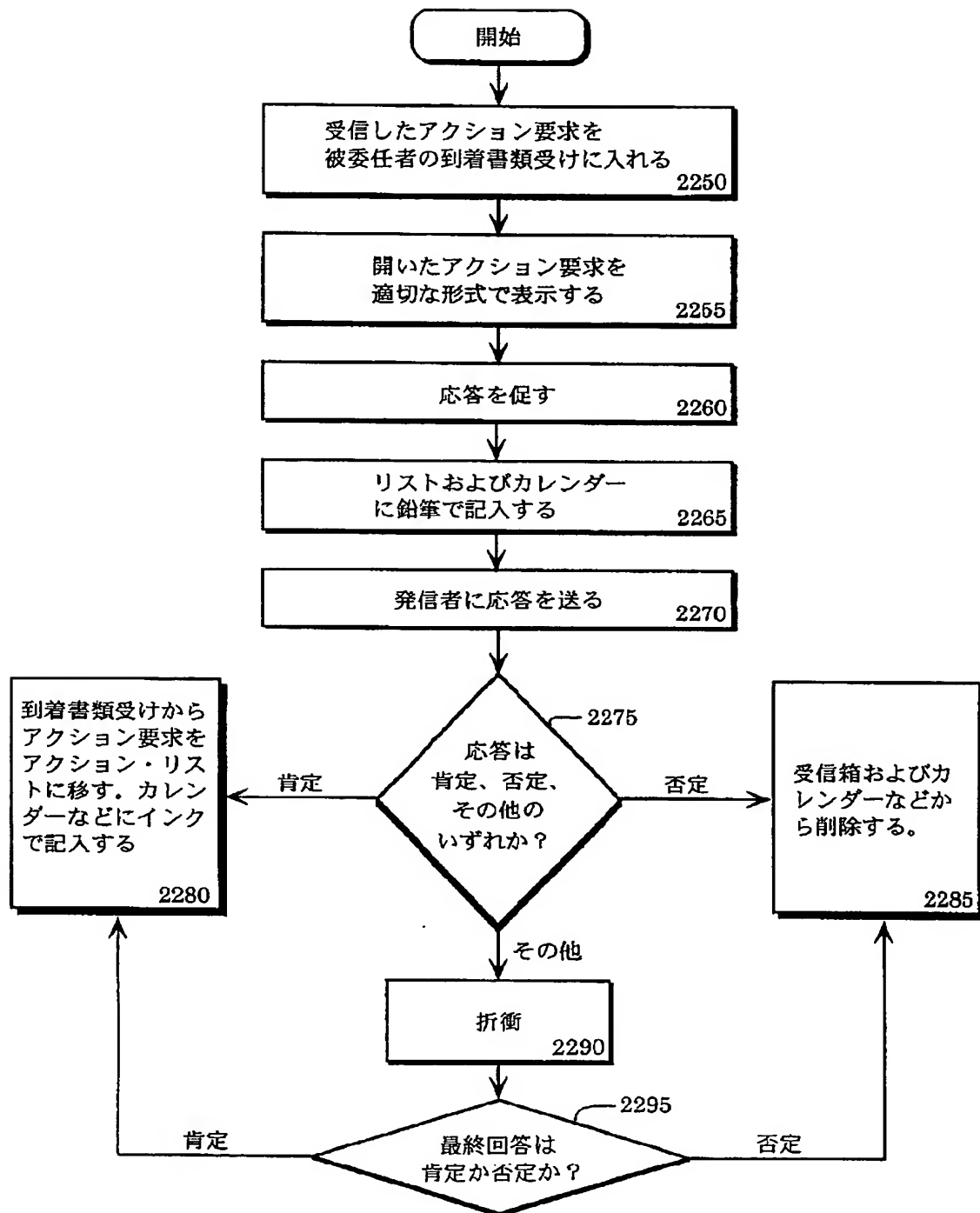


FIG. 22B

【図23】

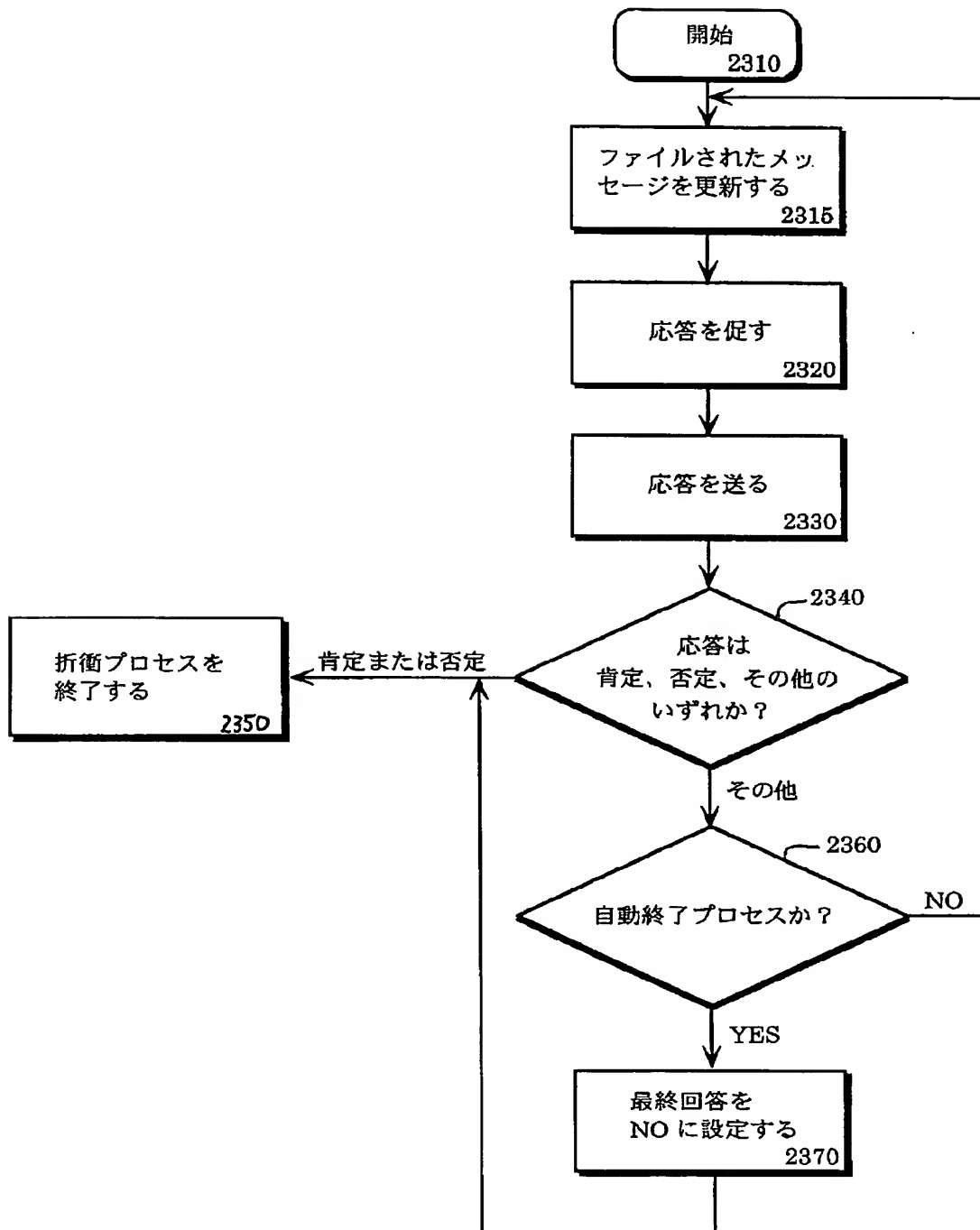
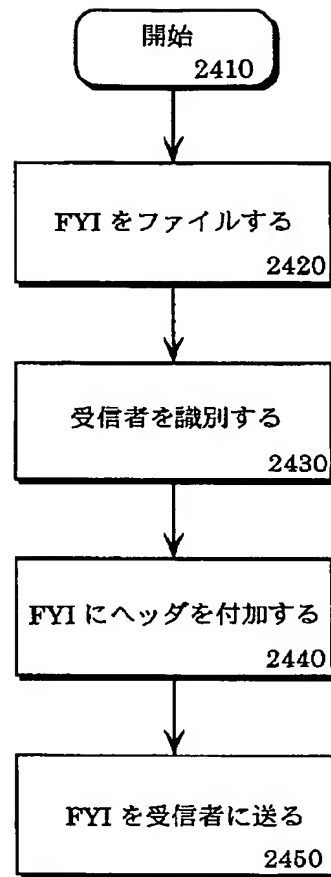


FIG. 23

【図 24】

**FIG. 24**

【図25】

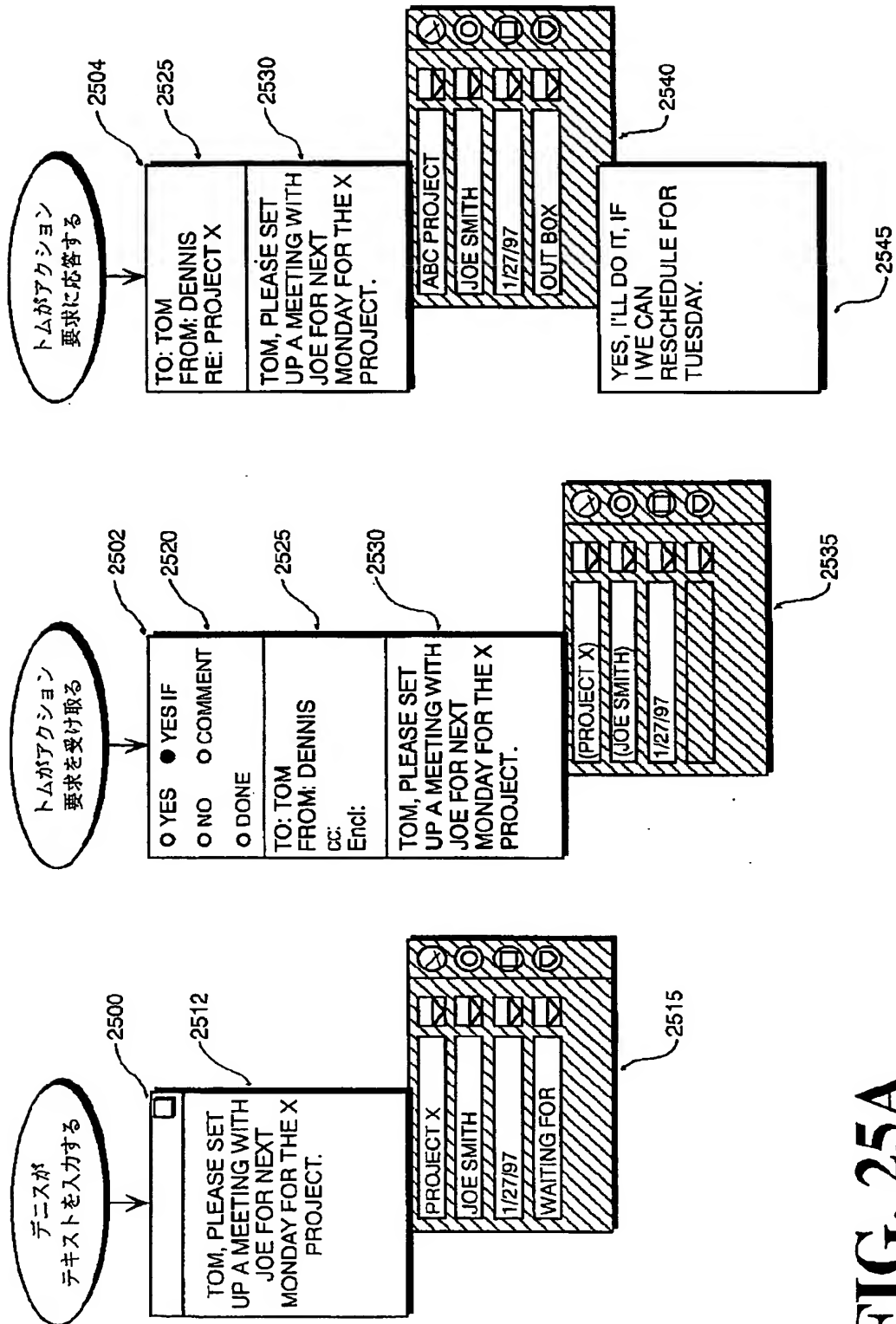


FIG. 25A

【図25】

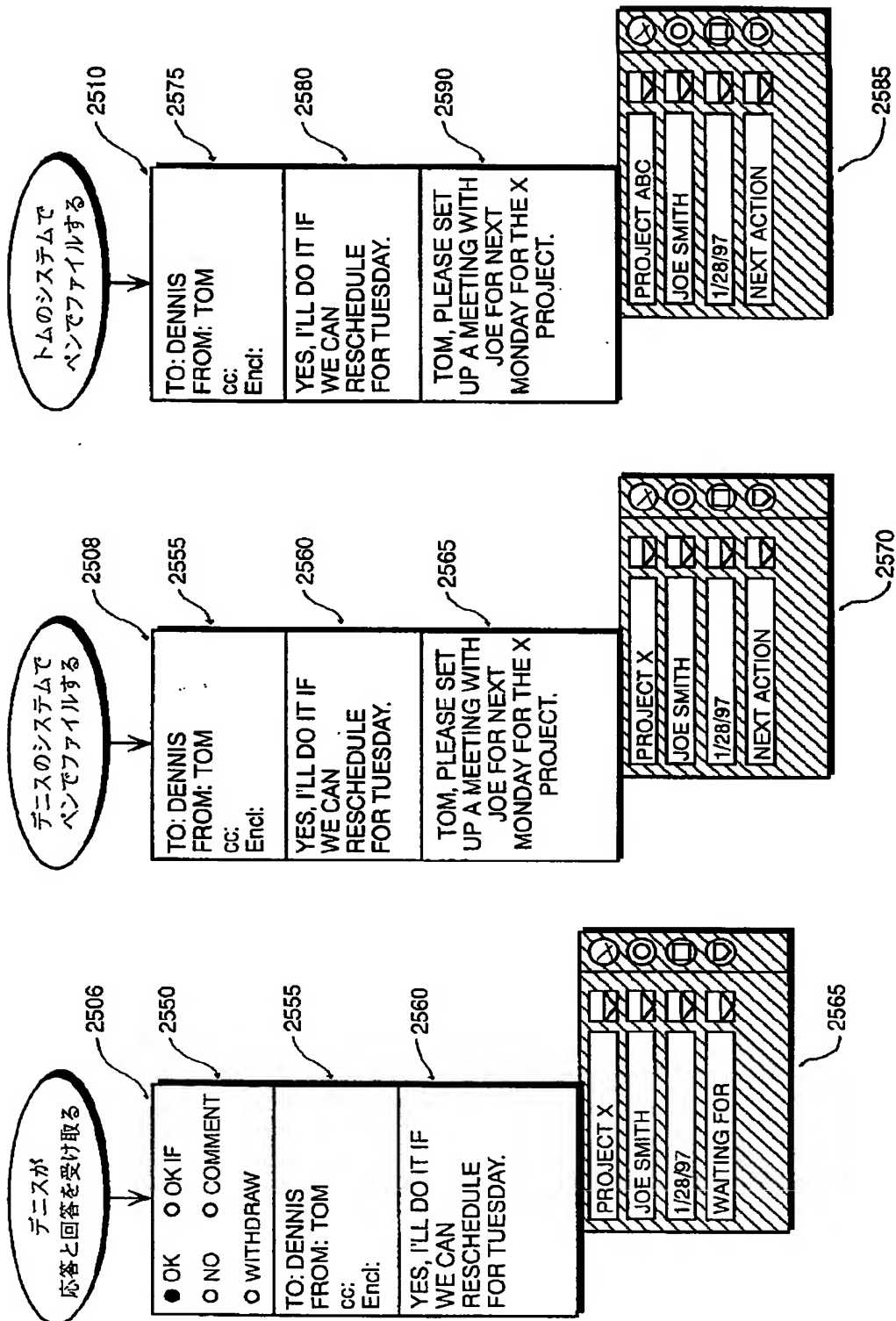


FIG. 25B

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/02217

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 17/30

US CL : 707/1.2

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/1.2

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

ACM web site, IEEE, Dialog

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US 5,634,121 A (TRACZ et al) 27 May 1997, col. 1, lines 23-63.	1
X,E	US 5,732,274 A (O'NEILL) 24 March 1998, col. 4, lines 52-63.	2

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document published on or after the international filing date

L document which may throw doubts on priority claim(s) in which is cited to establish the publication date of another citation or other special reason (as specified)

O documents referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

A document member of the same patent family

Date of the actual completion of the international search

09 JUNE 1998

Date of mailing of the international search report

14 AUG 1998

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

MICHAEL WALLACE

Telephone No. (703) 308-8996

フロントページの続き

(81)指定国 EP(AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW

(72)発明者 ラバニ, ダニー

アメリカ合衆国・94109・カリフォルニア
州・サンフランシスコ・チェストナット
ストリート・8番・1027

(72)発明者 ブックハイム, デニス

アメリカ合衆国・94309・カリフォルニア
州・パロアルト・アシュトンアヴェニ
ュ・570

(72)発明者 マン, トニー

アメリカ合衆国・98027・ワシントン州・
イサクア・サウスイースト164ティエ
イチストリート・25628

(72)発明者 ヘーガン, トーマス

アメリカ合衆国・02109・マサチューセッ
ツ州・ボストン・ユニオンワーフ・13

(72)発明者 ジョアクイン, ジェームズ

アメリカ合衆国・94117・カリフォルニア
州・サンフランシスコ・デュボーストリ
ート・4番・460

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.